Multigroup latent variable modelling with the Mplus software (V6)

Jouni Kuha

Department of Statistics and Department of Methodology
London School of Economics and Political Science

5 October 2013

Available at `http://stats.lse.ac.uk/lcat/`[1]

[1]This document was prepared as part of the research project *Latent variable modelling of categorical data: Tools of analysis for cross-national surveys (LCAT)*, funded by the Economic and Social Research Council under grant RES-239-25-0022. See the project website (`http://stats.lse.ac.uk/lcat/`) for more information.

Contents

1 Introduction **4**

1.1 Mplus language 4

1.2 Data definition and setup 5

1.2.1 Transferring data from other programs 5

1.2.2 DATA command 7

1.2.3 VARIABLE command 8

1.2.4 DEFINE command 10

2 Multigroup latent variable models: General specification **11**

3 Linear factor analysis models for multiple groups **14**

3.1 General specification of the models 14

3.2 MODEL command 15

3.3 1-factor multigroup models in Mplus 18

3.3.1 Input 18

3.3.2 Output 24

3.4 2-factor multigroup models in Mplus 27

3.4.1 Input 27

3.4.2 Output 32

4 Latent trait models for multiple groups **36**

4.1 General specification of the models 36

4.2 1-trait multigroup models in Mplus 39

4.2.1 Input 39

4.2.2 Output 48

4.3 2-trait multigroup models in Mplus 52

4.3.1 Input 52

4.3.2 Output 57

5 Latent class models for multiple groups **61**

5.1 General specification of the models 61

5.2 Mplus input 62

5.3 Mplus output 69

6 Other topics **73**

6.1 Estimation of the models 73

6.2 Model assessment statistics 73

7 Using R software to work with Mplus **75**

7.1 Installation 75

7.2 Basic ideas of the `lcat` functions 75

7.3 Uploading models: The `lcat` main function 76

7.3.1 Setup 76

7.3.2 Referring to paths and files 77

7.3.3 Running models from R 77

7.3.4 Loading results into R 77

7.4 Looking at the results: `print` 78

7.5 Tidying up the results: `reorder` 78

7.6 Plots of item response probabilities: `plot` 80

7.7 Likelihood ratio tests: `lcat.lrtest` 81

7.8 Residuals: `resid` 81

A Data set used for the examples **83**

B `MplusAutomation` package in R **84**

B.1 Creating input data and command files 84

B.2 Running models in Mplus 85

B.3 Returning Mplus results to R 85

C Syntax of `lcat` functions **86**

C.1 Main function `lcat` 86

C.2 Residuals function `resid` (i.e. `residuals.lcat`) 87

C.3 Likelihood ratio test function `lcat.lrtest` 88

C.4 Post-processing function `reorder` (i.e. `reorder.lcat.list`) 88

C.5 Printing function `print` (i.e. `print.lcat.list`) 90

C.6 Plotting function `plot` (i.e. `plot.lcat.list`) 91

1 Introduction

This document describes how some common types of latent variable models can be estimated with the Mplus software[2]. The focus is on multigroup models, i.e. models which have a single categorical explanatory variable. More information can be found in the Mplus user's guide (Muthén and Muthén 2007) and technical appendices (Muthén 2004). See standard references for more information on latent variable modelling in general. The LCAT website (`http://stats.lse.ac.uk/lcat/`) contains the data sets and input and output files for most of the examples discussed here.

This first section covers the basics of the Mplus syntax, and the initial commands DATA, VARIABLE and DEFINE which are used to set up data for analysis.

1.1 Mplus language

An Mplus analysis is specified by a set of *commands* and their *options* specified in a *syntax file*, which is a standard text (ascii) file with the default file name extension `.inp`. See Section 19 of the manual for a complete listing of the language. A model is fitted by executing the commands in a syntax file, typically from within the Mplus Editor.

Comments are indicated by an exclamation point (!). Anything on after a ! on a line is ignored by the program.

The Mplus language consists of ten **commands**:

- TITLE:
- DATA: (compulsory)
- VARIABLE: (compulsory)
- DEFINE:
- ANALYSIS:
- MODEL:
- OUTPUT:
- SAVEDATA:
- PLOT:
- MONTECARLO:

The TITLE command is followed by user-specified text which appears at the top of the output as a title for the analysis. This can extend over several lines. The other commands are described in more detail below.

[2]Mplus version 6.12 was used to test the examples in this document.

The DATA and VARIABLE commands must be included in every analysis, the others are optional. Each command must begin on a new line, and the name of the command must be followed immediately by a colon (e.g. DATA:). The commands can be in any order, but it is sensible to use them roughly in the order shown above.

Each command (apart from TITLE) has several **options**. Most have default values or are optional, so they need not always be used. The value(s) of an option are set by statements of the form

```
<OPTION> = <VALUES>;
```

listed under the command to which the options belong. Note the semicolon (;) at the end of the statement. If there are several elements in `<VALUES>`, these can be separated by blanks or commas. In most cases, `ARE` and `IS` can be used instead of the `=` above. In this document, only `=` is used.

A hyphen (-) can be used to shorten lists of variables or numbers. For example, the command

```
VARIABLE: NAMES = y1-y4;
```

declares that the 4 variables in the input data set will be named (in order of appearance in the file) `y1, y2, y3, y4`. Note that hyphened lists of variables used subsequently are read in order of declaration of the variable names, not (e.g.) numerical or alphabetical order. For example, in

```
VARIABLE: NAMES = y1 y2 y4 y3 y5 y6;
MODEL: factor BY y1 - y4;
```

`factor` is measured by `y1, y2, y4`, *not* by `y1, y2, y3, y4`. Many options also allow the value `ALL`, which refers to all variables (in a sense appropriate for the context of use).

Mplus is not case-sensitive about the names of commands, options or variables. Names of options can be shortened to four or more letters, and values of options to the letters shown in bold in Section 19 of the manual.

1.2 Data definition and setup

1.2.1 Transferring data from other programs

Mplus reads in data from an external text (ascii) file, which must have a very simple format:

- Only numerical data, with the possible exception of a single non-numeric missing-value code (see Section 1.2.3 below).
- In "free format" data, each entry is separated by a comma, space or tab, and blanks for missing data are not allowed. (A fixed format is also possible, but not discussed here.)
- No variable names on the first row of the file.

As an example, consider a data set from Round 4 of the European Social Survey which will be used for all the examples in this document. The data set and variables are described in Appendix A on p. 83.

The first step is to export the file as an ascii file from other software. Here we consider three common statistical packages, SPSS, Stata, and R. In each case the result is a text file called `ess4_3c.dat`.

SPSS

SPSS syntax for exporting the file is of this form:

```
SAVE TRANSLATE OUTFILE='D:\LCAT\ess4_3c.dat'
  /TYPE=CSV
  /MAP
  /REPLACE
  /KEEP=idno cntrynum ppltrst pplfair pplhlp
         polintr polcmpl poldcs trstprl trstlgl trstplc trstplt trstprt.
```

This file is saved in a comma-delimited form.

Stata

For Stata, there is an add-on package `stata2mplus` (developed by Michael Mitchell) which creates both the ascii data file and a basic Mplus input file corresponding to it. An example is

```
stata2mplus idno cntrynum ppltrst-trstprt ///
        using d:\lcat\ess4_3c, replace missing(99)
```

This creates the comma-delimited data file `ess4_3c.dat`, where all missing values are coded as 99. The command also creates a basic input file `ess4_3c.inp` (see below) in the same directory where the data file is saved.

There is also a more general Stata add-on package called `runmplus` (developed by Richard Jones) which calls Mplus from within Stata and returns the results back to Stata. From the user's point of view, this in effect turns Mplus into a Stata procedure where the Mplus commands are entered in Stata as options to the `runmplus` command.

R

Standard R commands (e.g. `write.csv`) can be used to export data to a text file. There is also the function `prepareMplusData` in the package `MplusAutomation` (developed by Michael Hallquist and Joshua Wiley), which also prints corresponding syntax for a basic Mplus input file on the R console. Other facilities of this package are discussed further in Appendix B.

```
prepareMplusData(ess4.3c.dat, file="d:/lcat/ess4_3c.dat",
   keepCols=c("idno","cntrynum","ppltrst","pplfair","pplhlp",
              "polintr","polcmpl","poldcs",
              "trstprl","trstlgl","trstplc","trstplt","trstprt"))
```

The saved file is tab-delimited, although this should not concern the user. Mplus syntax for reading in the data is the same for both comma-delimited and tab-delimited input files.

Below is an Mplus input file which reads in the data file `ess4_3c.dat` created by any of these methods, and produces summary statistics. The commands used here are discussed in more detail below. Note already that the example illustrates different ways of specifying codes for missing observations, which may depend on which software was used to export the data file. Here the missing-value code is specified as 99 for all the variables, as is the case for the Stata example above. The commented-out lines illustrate two other cases: one where different missing-value codes are used for different variables (as when these data are exported from SPSS) and one where the code is a full stop (.) for all observations (as when exporting these data from R).

```
Title: LCAT examples
    Reading in data + summary statistics
Data:
  File = ess4_3c.dat ;
Variable:
  Names =
     idno cntrynum ppltrst pplfair pplhlp polintr polcmpl poldcs trstprl
     trstlgl trstplc trstplt trstprt;
  Missing = all (99) ; ! Missing-value code in file exported from Stata is 99
! Examples of specifications of other missing-value codes for this data set:
! Exported from SPSS:
! Missing = ppltrst-pplhlp (66-99) polintr-poldcs(6-9) trstprl-trstprt(66-99);
! Exported from R:
! Missing = all .;
Analysis:
  Type = basic ;
```

1.2.2 DATA command

The DATA command defines the data set used for the analysis. Its most important option is

- `FILE = <filename>;`, e.g. `FILE = d:\lcat\ess4_3c.dat;`. This reads in a data set from a file. For the format of the file, see S. 1.2.1 above. If the path or filename contains spaces, the whole path and filename must be in quotes. If the full path is not given, Mplus looks for the file in the local directory, which usually the directory from which the most recent input file was opened.

There are also six "DATA transformation commands". An example of is `DATA MISSING:`. It creates binary missingness indicators (with value 1 if an observation is missing, 0 if not) corresponding to named variables in the data. Its two options are (with example variable names):

- `NAMES = y1-y4;` Create missingness indicators for variables y1–y4.
- `BINARY = d1-d4;` The new missingness indicators will be named d1–d4 respectively.

1.2.3 VARIABLE command

The VARIABLE command does most of the work in specifying the contents of the data set. It has a large number of options. Below we describe only the most important of them:

- `NAMES`: assigns names to *all* the variables in the input data set, in the order they appear in the data set. Names can be up to 8 characters long, can include only letters, numbers and the underscore (_), and must begin with a letter. Lists are allowed in this declaration, e.g. `NAMES = y1-y3` means `y1, y2, y3` and `ya-yc` means `ya, yb, yc`.
- `USEOBSERVATIONS`: Selects observations to be included in the analysis. The value of this option is a logical expression for the variables in declared by `NAMES`. The logical operators in Mplus are `AND, OR, NOT, ==, /=, >=, <=, >, <`. For example:
    - `USEOBSERVATIONS = x1 < 10 AND sex==1;`
- `USEVARIABLES`: Selects variables to be included in the analysis. This is a list of variable names. The names refer to variables declared by the `NAMES` option ("old variables" below) or defined by DATA transformation commands or the DEFINE command (see below; "new variables"). The *order* of the variables matters here:
    - all old variables must be listed before all new variables; within these categories, the order is free
    - this option redefines the order of the variables implied by subsequent lists of variables, *after* the VARIABLE command (but not yet for other options of that command); e.g. if `USEVARIABLES = y1 y3 y2 y4`, subsequently the list `y1-y2` within (say) an ANALYSIS command means `y1, y3, y2`
    - old variables which have special functions and are declared elsewhere (e.g. by `GROUPING` or `IDVARIABLE`) need not be listed here, but *new* variables used in these roles must be declared also here
- `MISSING`: specifies the missing value codes. This can be either a single non-numerical code or one or more numerical codes, but not a mixture of the two. The non-numerical codes allowed are
    - period (`MISSING = .`) or asterisk (`MISSING = *`) [or, in fxed-format data only, a blank]

  Specification of numerical missing data codes allows both several different values and different codes for different variables. Examples:
    - `MISSING = ALL (99);`
    - `MISSING = ALL (9 99);`
    - `MISSING = y1 (9 95-99) y2-y5 (-9,99);`

  The codes must be separated by commas if any one of them is negative.
- Options for declaring types of dependent variables: `CATEGORICAL` (meaning ordinal), `NOMINAL`, `COUNT`, and `CENSORED`. The value of each of these is a list of (old or new) variables. Any observed variable not listed under one of these and used as a dependent

variable will be treated as continuous. Below I discuss the "categorical" and "nominal" types.

  - `CATEGORICAL`: variables which will be treated as *ordinal*. Typically binary variables are also listed here rather than as "nominal". A variable like this can have at most 10 categories.
    * estimation of the model for a variable like this codes the categories of the variable as 0,1,2,... in rank order of the values of the variable observed in the data set. However, for calculations in the DEFINE command, conditions in the USEOBSERVATIONS option etc., the original values are used.
    * *thresholds* for the variable (intercept terms in a model for cumulative probabilities, see discussion in Section 4) are referred to with the symbol $. For example, for a four-category variable `y1` they are labelled `y1$1, y1$2, y1$3`.
  - `NOMINAL`: variables which will be treated as *nominal* (unordered). A variable like this can have at most 10 categories.
    * estimation of the model for a variable like this codes the categories of the variable as 0,1,2,... in rank order of the values of the variable observed in the data set.
    * *categories* of the variable are referred to with the symbol #, and numbered 1,2,... The *last* (i.e. highest-numbered) category is used as the reference category and cannot be referred to. For example, for a four-category variable `y1` the first 3 categories are labelled `y1#1, y1#2, y1#3`.

- `CLASSES`: the core option for latent class analysis. It is used to specify the number and names of categorical latent variables, and the numbers of categories for each. Examples are
  - `CLASSES = class (4);`
  - `CLASSES = class (4) class2 (3);`

  where `class` and `class2` are latent variables, with 4 and 3 categories respectively.

- Some options identify variables which are not involved in the fitted model but which have other functions:
  - `IDVARIABLE`: name of an id variable which will not be used in the analysis but will be included in any individual-level data file saved by the SAVEDATA command
  - `AUXILIARY`: names of other variables which will not be used in the analysis but which will be included in any individual-level data file saved by the SAVEDATA command
    * This option also has a second use, to identify variables for which equality of means across latent classes is tested (using estimation via multiple imputation of the classes) after fitting a latent-class model. This possibility is not discussed in this document.
  - `CONSTRAINT`: Variables which are used in the MODEL CONSTRAINT command. This must include *all* such variables, including ones that are also used in the model.

1.2.4 DEFINE command

The DEFINE command is used to transform existing variables and to create new ones. The general format of conditional and unconditional transformations is

- `variable = mathematical expression;`
- `IF (<logical condition>) THEN variable = value;`

See pp. 466–468 of the manual for the syntax of such expressions, which is fairly standard. Note that for a logical condition the variable being assigned a value must exist already, created by using an earlier DEFINE command if necessary. For example:

```
VARIABLE:
    NAMES = y1-y3;
    USEVARIABLES = y1-y3 a b c;
DEFINE:
    a = 10*y1;
    b = a;
    IF (y1>0) THEN b = 3*y2;
    c = y3;
    CUT c(0);
```

Note that to assign a missing value to an observation, the keyword _MISSING is used.

An additional useful option for DEFINE is CUT, which is used to categorise continuous variables. More than one CUT statement can be included, and each CUT statement can refer to a single variable or a list of variables. The new values overwrite the original ones, so it is again necessary to apply the command to a copy of a variable if we want to retain also the original values. For example, the CUT command above redefines variable `c` to have two categories for `y3`≤ 0 and `y3`> 0 respectively. Similarly, `CUT y1-y3(-1 1)` would redefine each of `y1`, `y2` and `y3` to have 3 categories, corresponding to the original values of these variables being ≤ -1, $(-1, 1]$ and > 1 respectively. The new categories are coded (e.g. if the data are saved using the SAVEDATA command) $0, 1, \ldots$, even though in Mplus output they are called $1, 2 \ldots$.

2 Multigroup latent variable models: General specification

In this section we leave Mplus for a moment, to define the class of models we consider in more general terms. Consider vectors of three types of variables:

- observed *covariates* $\mathbf{x}$,
- *latent variables* $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_q)'$, and
- observed measurements or *items* $\mathbf{y}$

for a single *unit* such as a respondent in a survey. We consider models of the general form

$$p(\mathbf{y}, \boldsymbol{\eta}|\mathbf{x}) = p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{x})\, p(\boldsymbol{\eta}|\mathbf{x}) \tag{1}$$

where $p(\cdot|\cdot)$ denotes a conditional density function. We refer to $p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{x})$ as the *measurement model* and $p(\boldsymbol{\eta}|\mathbf{x})$ as the *structural model.*

Estimation of models is based on the conditional density

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{x})\, p(\boldsymbol{\eta}|\mathbf{x})\, d\boldsymbol{\eta}. \tag{2}$$

We assume here that the values of $\mathbf{y}_i$ given $\mathbf{x}_i$ are independent across units i, so the likelihood function will be the product $\prod_i p(\mathbf{y}_i|\mathbf{x}_i)$. Taking this as given, we consider below expressions like (1) and (2) for one unit and omit the subscript pertaining to the unit.

Often the measurement models are such that each variable in $\mathbf{y}$ measures directly only one latent variable. We can then write $\mathbf{y} = (\mathbf{y}_1', \ldots, \mathbf{y}_q')'$, where $\mathbf{y}_k = (y_{k1}, \ldots, y_{kp_k})'$ is a vector of the p_k observed variables which are regarded as measurements of η_k $(k = 1, \ldots, q)$. The total number of items in $\mathbf{y}$ is then $p = \sum_k p_k$. We will most often assume that, conditional on η_k and $\mathbf{x}$, the measurements $\mathbf{y}_k$ are independent of all other $\eta_{k'}$, $k' \neq k$, and elements of $\mathbf{y}_k$ are independent of each other. The measurement model is then of the form

$$p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{x}) = \prod_{k=1}^{q} p(\mathbf{y}_k|\eta_k, \mathbf{x}) = \prod_{k=1}^{q} \prod_{j=1}^{p_k} p(y_{kj}|\eta_k, \mathbf{x}). \tag{3}$$

More general models are of course possible, i.e. ones where some items in $\mathbf{y}$ are measures of several of the latent variables in $\boldsymbol{\eta}$, or are not conditionally independent given $\boldsymbol{\eta}$. However, in our examples we will assume (3) unless otherwise stated.

We consider here *multigroup models* where the only covariate is a single categorical variable with G categories or *groups.* In the applications of cross-national survey analysis which motivated this work, the groups are typically countries of survey respondents. In the multigroup situation, $\mathbf{x}$ consists of indicator variables for $G-1$ of the groups, say $\mathbf{x} = (x_2, \ldots, x_G)'$ where x_g is an indicator for group g. Dependence on $\mathbf{x}$ then implies that a distribution varies across the groups. Substantive interest usually focuses on the structural model $p(\boldsymbol{\eta}|\mathbf{x})$, which shows how the distribution of the latent variables $\boldsymbol{\eta}$ varies across the groups. Any dependence on $\mathbf{x}$ in the measurement model $p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{x})$ indicates lack of *measurement equivalence*, i.e. that the measurement properties of at least one item vary across the groups, even conditional on the true value of $\boldsymbol{\eta}$.

In the multigroup context it is often more convenient to write (1) in the equivalent form

$$p^{(g)}(\mathbf{y}, \boldsymbol{\eta}) = p^{(g)}(\mathbf{y}|\boldsymbol{\eta})\, p^{(g)}(\boldsymbol{\eta}), \quad g = 1, \ldots, G, \tag{4}$$

and the measurement model (3) as

$$p^{(g)}(\mathbf{y}|\boldsymbol{\eta}) = \prod_{k=1}^{q} p^{(g)}(\mathbf{y}_k|\eta_k) = \prod_{k=1}^{q} \prod_{j=1}^{p_k} p^{(g)}(y_{kj}|\eta_k) \tag{5}$$

where the superscript (g) indicates that a distribution pertains to group g. We refer to (4) as the "multiple-group specification" of a model, and (1) as the "covariate specification". In Mplus a particular way of specifying a multigroup model often corresponds more naturally to one than the other of these specifications.

For illustration, we will mostly consider in detail two simple special cases of (1)–(3):

- One latent variable ($q = 1$), so that the notation can be simplified to $\boldsymbol{\eta} = \eta$ and $\mathbf{y} = (y_1, \ldots, y_p)'$, and the model is

$$p(\mathbf{y}, \eta|\mathbf{x}) = \left[\prod_{j=1}^{p} p(y_j|\eta, \mathbf{x})\right] p(\eta|\mathbf{x}) \tag{6}$$

  or, in the multiple-group notation,

$$p^{(g)}(\mathbf{y}, \eta) = \left[\prod_{j=1}^{p} p^{(g)}(y_j|\eta)\right] p^{(g)}(\eta) \tag{7}$$

- Two latent variables $\boldsymbol{\eta} = (\eta_1, \eta_2)'$, so that

$$\begin{aligned} p(\mathbf{y}, \boldsymbol{\eta}|\mathbf{x}) &= \left[\prod_{j=1}^{p_2} p(y_{2j}|\eta_2, \mathbf{x})\right] \left[\prod_{j=1}^{p_1} p(y_{1j}|\eta_1, \mathbf{x})\right] p(\eta_2, \eta_1|\mathbf{x}), \quad \text{i.e.} &(8)\\ p^{(g)}(\mathbf{y}, \boldsymbol{\eta}) &= \left[\prod_{j=1}^{p_2} p^{(g)}(y_{2j}|\eta_2)\right] \left[\prod_{j=1}^{p_1} p^{(g)}(y_{1j}|\eta_1)\right] p^{(g)}(\eta_2, \eta_1). &(9)\end{aligned}$$

  In (8) and (9) the two latent variables are treated on an equal footing. We will also consider formulations where η_1 is treated as a predictor of η_2, in which case we can further write

$$\begin{aligned} p(\eta_2, \eta_1|\mathbf{x}) &= p(\eta_2|\eta_1, \mathbf{x})\, p(\eta_1|\mathbf{x}) \qquad \text{i.e.} &(10)\\ p^{(g)}(\eta_2, \eta_1) &= p^{(g)}(\eta_2|\eta_1)\, p^{(g)}(\eta_1). &(11)\end{aligned}$$

Path diagrams for these basic cases are shown in Figure 1. Different types of multiple-group latent variable models are obtained with different choices for the distributions $p^{(g)}(\mathbf{y}|\boldsymbol{\eta})$ and $p^{(g)}(\boldsymbol{\eta})$ in these general formulations. In the sections below we consider three such special cases, and how they can be implemented in Mplus.

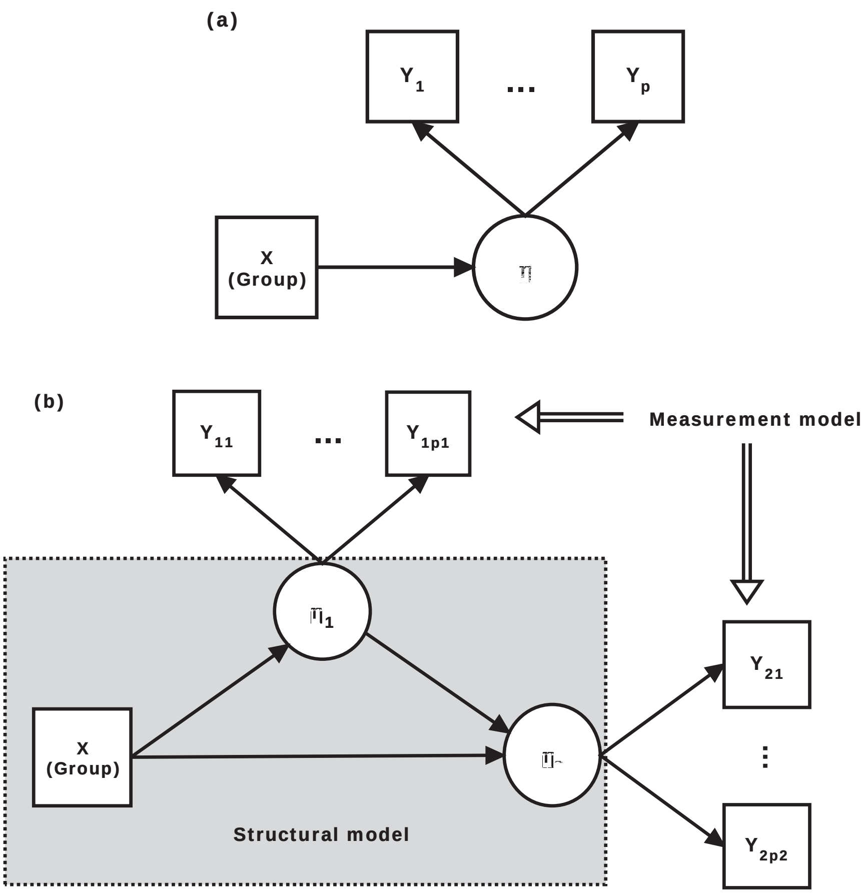

Figure 1: Two basic multiple-group latent variable models, with (a) one and (b) two latent variables.

3 Linear factor analysis models for multiple groups

3.1 General specification of the models

Here we us the term *linear factor analysis* to refer to both (confirmatory) *factor analysis* and *structural equation modelling* in conventional terminology. Here both the observed indicators $\mathbf{y}$ and the latent variables (*factors*) $\boldsymbol{\eta}$ are taken to be continuous variables and modelled using normal linear regression models.

Consider first the one-factor model, focusing on the multiple-group formulation (7). Here in group $g = 1, \dots, G$, the measurement model for an item y_j, $j = 1, \dots, p$, is

$$y_j^{(g)} = \tau_j^{(g)} + \lambda_j^{(g)} \eta^{(g)} + \epsilon_j^{(g)} \tag{12}$$

where $\epsilon_j \sim N(0, \theta_j^{(g)})$, and the structural model for the factor η is

$$\eta^{(g)} \sim N(\kappa^{(g)}, \phi^{(g)}) \tag{13}$$

where the superscript (g) is added also to the random variables y_j, η and ϵ_j to make it clear which groups the observations belong to. Here $\tau_j^{(g)}$ (*intercepts*), $\lambda_j^{(g)}$ (*factors loadings*) and $\theta_j^{(g)}$ (*error variances*) are the unknown parameters of the measurement model for item y_j, and $\kappa^{(g)}$ (*factor means*) and $\phi^{(g)}$ (*factor variances*) are the parameters of the structural model, i.e. the distribution of the factor η.

In (7), the observed items are assumed conditionally independent given the latent variable, i.e. $\text{cov}(\epsilon_j^{(g)}, \epsilon_{j'}^{(g)}) = \theta_{jj'} = 0$ for all $j \neq j'$. This is not essential, so error covariances between some pairs of items can be included, as long as there are few enough of them so that the model remains identifiable. We include a few such examples below.

The model has complete *measurement equivalence* (or "invariance") across the groups if each of the parameters of the measurement model has the same value in all of the groups, i.e. if $\tau_j^{(1)} = \cdots = \tau_j^{(G)}$, $\lambda_j^{(1)} = \cdots = \lambda_j^{(G)}$ and $\theta_j^{(1)} = \cdots = \theta_j^{(G)}$ for all $j = 1, \dots, p$. When this is the case for a parameter, we omit the supercript (g) and write $\tau_j^{(g)} = \tau_j$, $\lambda_j^{(g)} = \lambda_j$ or $\theta_j^{(g)} = \theta_j$ for all j.

Even when the model specifies complete measurement equivalence, some constraints must be set for the parameters to ensure identifiability of the scale of the latent variable. Two alternative constraints are commonly used. The first is to fix the intercept and loading of one item at known values, usually 0 and 1 respectively, i.e. $\tau_j = 0$ and $\lambda_j = 1$ for one j, say $j = 1$. The second is to fix the mean and variance of the latent factor for one group, typically at 0 and 1 respectively, i.e. to set $\kappa^{(g)} = 0$ and $\phi^{(g)} = 1$ for one g, say $g = 1$. In a multigroup analysis the constraint on the distribution of the latent variable is preferable, because it clearly separates the condition required to identify the latent scale from constraints on the measurement parameters which are used to specify different levels of measurement equivalence and non-equivalence. In the examples below we always use the constraints $(\kappa^{(1)}, \phi^{(1)}) = (0, 1)$, except in one example (Model E2B) which is included to show how the constraint $(\tau_1, \lambda_1) = (0, 1)$ is implemented in Mplus.

Consider now the linear factor analysis model with two latent factors $\boldsymbol{\eta} = (\eta_1, \eta_2)$. In general, the measurement model for an item y_j in group g is

$$y_j^{(g)} = \tau_j^{(g)} + \lambda_{1j}^{(g)} \eta_1^{(g)} + \lambda_{2j}^{(g)} \eta_2^{(g)} + \epsilon_j^{(g)} \tag{14}$$

where $\epsilon_j \sim N(0, \theta_j^{(g)})$. As discussed above, in our examples we assume throughout that each item measures only one factor, so that $\lambda_{1j}^{(g)} = 0$ or $\lambda_{2j}^{(g)} = 0$ for every j; however, this assumption can of course be easily relaxed.

The new aspect of models with two or more factors is the increased richness of the structural model for the factors, and how this may vary across groups. Note first that the structural model may be equivalently specified with the factors treated on an equal footing as in (9) and their association specified in terms of a covariance (called "covariance specification" in our examples), or as a regression model ("regression specification") with one factor treated as explanatory to the other, as in (11). We will give examples of both of these. In real analysis the choice will depend on the research questions and other substantive considerations.

In the covariance specification, the structural model specifies that the two factors η_1 and η_2 are jointly normally distributed with marginal distributions $\eta_1^{(g)} \sim N(\kappa_1^{(g)}, \phi_{11}^{(g)})$ and $\eta_2^{(g)} \sim N(\kappa_2^{(g)}, \phi_{22}^{(g)})$, and covariance $\text{cov}(\eta_1^{(g)}, \eta_2^{(g)}) = \phi_{12}^{(g)}$ in groups $g = 1, \ldots, G$. To identify the scale of the latent variables, we specify $(\kappa_1^{(1)}, \phi_{11}^{(1)}) = (\kappa_2^{(1)}, \phi_{22}^{(1)}) = (0, 1)$.

For the regression specification, suppose that we want to treat η_1 as an explanatory variable for η_2. The structural model then specifies that

$$\eta_1^{(g)} \sim N(\kappa_1^{(g)}, \phi_{11}^{(g)}) \quad \text{and} \tag{15}$$

$$\eta_2^{(g)} = \gamma_0^{(g)} + \gamma_1^{(g)} \eta_1^{(g)} + \zeta^{(g)} \tag{16}$$

where $\zeta^{(g)} \sim N(0, \psi^{(g)})$, independent of $\eta_1^{(g)}$. To identify the scales of the latent variables, it is sufficient to assume that $(\kappa_1^{(1)}, \phi_{11}^{(1)}) = (\gamma_0^{(1)}, \psi^{(1)}) = (0, 1)$. The regression specification is equivalent to the covariance specification with $\kappa_2^{(g)} = \gamma_0^{(g)} + \gamma_1^{(g)} \kappa_1^{(g)}$, $\phi_{22}^{(g)} = (\gamma_1^{(g)})^2 \, \phi_{11}^{(g)} + \psi^{(g)}$ and $\phi_{12}^{(g)} = \gamma_1^{(g)} \phi_{11}^{(g)}$. (Note that the identifiability constraints we use are not quite identical, since assuming $\phi_{11}^{(1)} = \psi^{(1)} = 1$ under the regression specification implies the constraint $\phi_{22}^{(1)} = (\gamma_1^{(g)})^2 + 1$ for the unconditional variance of $\eta_2^{(1)}$ under the covariance specification.)

Subject to the minimal identifiability constraint for means and variances of the latent variables in one group, the model may be modified in a number of ways. First, the structural model may include further equality constraints for the parameters of the distributions of the latent variables across groups. Second, measurement equivalence of some items may be relaxed. In Sections 3.3 and 3.4 we consider various examples of such models, and how they can be implemented in Mplus.

3.2 MODEL command

Specification of a model and its estimation in Mplus are done through the MODEL and ANALYSIS commands. These have very many options, which are not all covered here (see the Mplus manual and technical appendices for more information). For linear factor analysis, only the MODEL command is typically needed. Its basic elements are reviewed in this section. Most of these apply also in the contect of the other models considered later. The ANALYSIS command is also used for those models, so relevant uses of it are discussed later.

Linear factor analysis is the default in Mplus: If observed variables are not defined as "categorical" or "nominal" in the VARIABLES command, and if latent variables are not defined as categorical (through the ANALYSIS: TYPE=MIXTURE option as discussed later), all of them are assumed to be continuous and modelled using linear models.

In the MODEL command, statements involving the keywords BY and ON define the regression relationships:

- BY defines a measurement model such as (12).
- ON defines a regression model, usually a structural model such as (16). This may involve as explanatory variables both latent variables and observed covariates such as dummy variables for groups in a covariate specification of a multiple-group model ($\mathbf{x}$ in the notation of (1)). However, a multigroup model can also often be specified in ways which do not explicitly include the group dummies as covariates, i.e. using the multiple-group specification (4). The details of how this is done depend on the model type.

The practical distinction between these keywords is that the left-hand side of a BY statement *defines* and names latent variables, which thus do not need to have been declared before. In contrast, all variable names mentioned on the left-hand side of an ON statement, or on the right-hand side of either statement, must have been defined previously, either by the VARIABLE command or by an earlier BY statement.

For example, suppose y1–y8 are observed indicators, eta1 and eta2 are continuous latent variables and x is a single dummy variable for a group. Then a model specification might have the form

```
MODEL:
        eta1 BY y1-y4;
        eta2 BY y5-y8;
        eta2 ON eta1 x;
        eta1 ON x;
```

which is an example of the kind of model shown in plot (b) of Figure 1. Here y1–y8 and x must have been identified as observed variables in the VARIABLES command.

Key syntax tools for various specifications on the parameters are the following:

- *<starting value>: freeing parameters to be estimated (when they are otherwise fixed for identification) and (optionally) assigning starting values for them
- @<value>: fixing parameters at given values

- `(<number>)`: constraining parameter values to be equal; all parameters given the same number will be fixed to be equal
- `(<label>)`: assigning labels to parameters, which can then be referred to in the MODEL CONSTRAINT command and elsewhere. Parameters that are assigned the same label are constrained to be equal.

Different types of parameters for linear models are referred to in the following ways. In each case, an example of the use of the specifications above is given for illustration.

- Means and intercept terms in both structural and measurement models, such as the τ, κ and γ_0 parameters of the models of Section 3.1: square brackets `[]`
  - e.g. `[y1 y2 y3](1)` — intercepts of `y1`, `y2` and `y3` constrained to the equal
- Regression coefficients (loadings) in structural and measument models (λ and γ_1 parameters above): the `BY` and `ON` commands
  - e.g. `eta1 BY y1* y2-y3` — loading of `y1` free to be estimated (by Mplus default it would be fixed at 1)
  - e.g. `eta1 ON x1 (p1)` — coefficient of `x` in model for `eta1` assigned the label "`p1`"
- Variances and residual variances (θ, ϕ_j and ϕ_{jj} parameters above): just the variable name
  - e.g. `eta1@1` — variance of `eta1` (in the structural model) fixed at 1
- Covariances and residual covariances (ϕ_{12} and any non-zero covariances $\theta_{jj'}$): `WITH`
  - e.g. `y1 WITH y2*0.5` — residual covariance of `y1` and `y2` freed to be estimated, with a starting value 0.5

There are also convenience options PWITH and PON for shortening long lists of pairwise dependencies, and lists of variable names can also be used as usual; see the manual for examples.

Parameter constraints can also be imposed. Estimates of two or more parameters which are given the same label are constrained to be equal. More general linear and non-linear constraints can be imposed by the MODEL CONSTRAINT command. These can involve both variables, model parameters — referred to using labels defined in the MODEL command — and new parameters, defined in the MODEL CONSTRAINT command using the NEW option. Two simple examples are shown here, to give the flavour. Here it is assumed that `p1`, `p2` and `p3` are labels for parameters.

```
MODEL CONSTRAINT:
        p3 = p2**2 + p1**2;
MODEL CONSTRAINT:
        0 = exp(p2) + exp(p1);
```

3.3 1-factor multigroup models in Mplus

3.3.1 Input

Throughout this manual, a data set from Round 4 of the European Social Survey (ESS) is used to illustrate various models. The data are introduced in Appendix A. There are three countries (Belgium, Bulgaria and Cyprus) and 11 observed items. We first consider a one-factor model for 5 of the observed variables (*tparl* to *tparties*). These are all treated as continuous indicators of one latent variable, which we label *institutional trust*. This, like all the other examples below, is intended purely for illustration of the computations. Thus the selection of neither the countries nor of the variables is theoretically informed, and we do not even examine how well the models fit the data.

Recall that the 1-factor multigroup model in multiple-group formulation is given by

$$
\begin{aligned}
y_j^{(g)} &= \tau_j^{(g)} + \lambda_j^{(g)}\eta^{(g)} + \epsilon_j^{(g)} \quad \text{with} \quad \epsilon_j \sim N(0, \theta_j^{(g)}), && (17)\\
\eta^{(g)} &\sim N(\kappa^{(g)}, \phi^{(g)}) && (18)
\end{aligned}
$$

for items $j = 1, \ldots, p$ in groups $g = 1, \ldots, G$. We illustrate various variants of this model. Table 1 summarises these models and the key lines of the Mplus syntax for each of them. The full syntax for one of the models (model N2) is shown in Figure 2 below. Code for most of the other models is obtained from this input, by commenting and uncommenting lines from the syntax as discussed below. The full syntax for each of the examples is also available at the LCAT website, at `http://stats.lse.ac.uk/lcat/`.

For linear factor analysis models in Mplus we consider only syntax which implements the multiple-group formulation (4) of the model. A covariate specification would also be possible, by including dummy variables for the groups as explanatory variables **x**, but for linear factor models this has no relative advantage and is not considered.

Note first that we specify (with the exception of Model E2B, which illustrates the alternative constraint) the measurement model as

```
trust BY tparl* tlegal-tparties;
```

This assigns `trust` as the name of the latent factor. The `tparl*` indicates that the loading ($\lambda_1^{(g)}$) of the first indicator variable (`tparl`) will be estimated freely, rather than constrained at $\lambda_1^{(g)} = 1$ as is the Mplus default. Instead, the scale of the latent factor is identified by constraining it to be 1 in group 1 (here Belgium, i.e. $\kappa^{(1)} = 1$) with the command `trust@1;`.

The multiple-group specification is obtained with the GROUPING option of the VARIABLE command, in our example as

```
Grouping = country(1=Bel 2=Bul 4=Cyp);
```

Here `country` identifies the group variable in the input data set, 1, 2, and 4 are the values of it that correspond to the three groups that appear in the data, and `Bel`, `Bul` and `Cyp` are the labels assigned to these groups within the Mplus analysis.

Table 1: Summary of the 1-factor models considered in Section 3.3. See equations (17)–(18) for the notation, and Figure 2 for full input syntax for Model N2. The example involves three groups, `Bel` (the baseline group), `Bul` and `Cyp`.

	Key feature	Key lines in Mplus MODEL command (under `Model:` unless otherwise mentioned)
Models with measurement equivalence ($\tau_j^{(g)} = \tau_j$, $\lambda_j^{(g)} = \lambda_j$ and $\theta_j^{(g)} = \theta_j$ for all j): — to constrain θ_j, in all of these `Model: tparl-tparties (3-7);`		
E0	$\kappa^{(g)} = 0$, $\theta^{(g)} = 1$ for all g	`trust@1 (1); [trust@0] (2);`
E1	$\kappa^{(g)}$ varies, $\theta^{(g)} = 1$ for all g	`trust@1 (1);`
E2	both $\kappa^{(g)}$ and $\theta^{(g)}$ vary across g	`Model Bel: trust@1;`
E2B	Equivalent to E2, but with the identifiability constraint $\tau_1 = 0$, $\lambda_1 = 1$	`trust BY tparl tlegal-tparties;` `Model Bel: [trust*];`
E3	Like E2, but with one error correlation $\theta_{45}^{(g)} = \theta_{45} \neq 0$	`tpolitic WITH tparties (8);`
Models with partial non-equivalence of measurement for item 1 (`tparl`) *(some or all of $\tau_1^{(g)}$, $\lambda_1^{(g)}$ and $\theta_1^{(g)}$ may vary across g):*		
N1	$\theta_1^{(g)}$ vary	`tlegal-tparties (3-6);`
N2	$\theta_1^{(g)}$ and $\tau_1^{(g)}$ vary	`Model Bul: [tparl];` `Model Cyp: [tparl];` See Figure 2 for full syntax of N2.
N3	$\theta_1^{(g)}$, $\tau_1^{(g)}$ and $\lambda_1^{(g)}$ vary	`Model Bul: [tparl]; trust BY tparl;` `Model Cyp: [tparl]; trust BY tparl;`
N4	Like N3, but with one error correlation $\theta_{45}^{(g)} = \theta_{45} \neq 0$	`tpolitic WITH tparties (8);`
N5	Like N4, but $\theta_{45}^{(g)} \neq 0$ varies across g	`tpolitic WITH tparties;`
N6	1-factor model fitted separately for each group	See Figure 3.

In the MODEL command, there is one overall `Model:` entry and possibly separate ones for each of the groups (`Model Bel:`, `Model Bul:` and `Model Cyp:`). The latter may be used to request different parameter values in different groups, as shown below. Any specification which appears only under the overall `Model:` command and which is not modified under a group-specific command will apply similarly to all the groups.

First, we show four models with complete measurement equivalence, i.e. where $\tau_j^{(g)} = \tau_j$, $\lambda_j^{(g)} = \lambda_j$ and $\theta_j^{(g)} = \theta_j$ for all j. Mplus default is in fact to let the error variances $\theta_j^{(g)}$ vary freely across the groups. To constrain them to be equal across the groups, we use the command

```
Model:  tparl-tparties (3-7);
```

in all of Models E0–E3.

- **E0**: The factor means and variances are equal in all groups. This is achieved with the command options
  ```
  Model:
  ```

`trust@1 (1);` ($\phi^{(g)} = 1$ for all g)
`[trust@0] (2);` ($\kappa^{(g)} = 0$ for all g)

- **E1**: The factor means vary across groups, but the variances do not:
  `Model:`
  `trust@1 (1);` ($\phi^{(g)} = 1$ for all g)
  Since the latent means are not mentioned at all, Mplus uses the default specification where $\kappa^{(1)} = 0$ and $\kappa^{(g)}$ is unconstrained for $g > 1$.

- **E2**: Both factor means and variances vary across groups:
  `Model Bel:`
  `trust@1;` (to get the identifiability constraint $\phi^{(1)} = 1$)
  The identifiability constraint $\kappa^{(1)} = 0$ is specified by default.

- **E2B**: Equivalent to E2, but uses the alternative identifiability constraint $(\tau_1, \lambda_1) = (0, 1)$ instead of $(\kappa^{(1)}, \phi^{(1)}) = (0, 1)$:
  `Model:`
  `trust BY tparl tlegal-tparties;` (`tparl` instead of `tparl*` invokes the default that $\lambda_1 = 1$)
  `Model Bel:`
  `[trust*];` (to free $\kappa^{(1)}$)
  Since $\lambda_1 = 1$ fixed, $\theta^{(1)}$ is by default estimated freely. Similarly, since $\kappa^{(1)}$ is freed, $\tau_1 = 0$ is then by default constrained in turn.

- **E3**: Like E2, plus one non-zero error covariance $\text{cov}(\epsilon_4^{(g)}, \epsilon_5^{(g)}) = \theta_{45}^{(g)} = \theta_{45} \neq 0$, constrained to be equal across the groups:
  `Model:`
  `tpolitic WITH tparties (8);`

Next, we show five models where some or all of the measurement parameters for one item (`tparl`, which we will call item 1) may be different across groups g. In each case, all other measurement parameters are equal across groups, and factor means $\kappa^{(g)}$ and variances $\phi^{(g)}$ vary across groups.

- **N1**: Error variance $\theta_1^{(g)}$ varies across groups:
  `Model:`
  `tlegal-tparties (3-6);`
  Since `tparl` is not mentioned in this, $\theta_1^{(g)}$ will vary by default.

- **N2**: Error variance $\theta_1^{(g)}$ and intercept $\tau_1^{(g)}$ vary across groups:
  ```
  Model:
  tlegal-tparties (3-6);
  Model Bul:
  [tparl];
  Model Cyp:
  [tparl];
  ```
  The idea here is that since the intercept `[tparl]` is mentioned separately under the models for groups `Bul` and `Cyp`, this parameter is estimated separately for each of these (and separately from the intercept in the remaining group, `Bel`).

- **N3**: Error variance $\theta_1^{(g)}$, intercept $\tau_1^{(g)}$ and loading $\lambda_1^{(g)}$ vary across groups:
  ```
  Model:
  tlegal-tparties (3-6);
  Model Bul:
  [tparl]; trust BY tparl;
  Model Cyp:
  [tparl]; trust BY tparl;
  ```
  with the same logic as in N2.

- **N4**: Like N3, plus one non-zero error covariance $\theta_{45}^{(g)} = \theta_{45} \neq 0$, constrained to be equal across the groups. The following line (which is the same as for E3) is added to N3:
  ```
  Model:
  tpolitic WITH tparties (8);
  ```

- **N5**: Like N4, but $\theta_{45}^{(g)} \neq 0$ varies across groups. The following line is added to N3:
  ```
  Model:
  tpolitic WITH tparties;
  ```
  Because $\theta_{45}^{(g)}$ is not explicitly constrained to be the same across groups, it varies by default. Note that in N5, letting only an error correlation vary across groups would not often make a great deal of sense. Both models N4 and N5 are included here purely to illustrate how models with different specifications for error covariances would be specified in Mplus.

- **N6**: This model is somewhat different from the others, and the syntax for it is shown separately in Figure 3 to avoid cluttering Figure 2 with too many commented-out lines. This is a model where *all* parameters vary freely across the groups. The result will be the same as if we fitted the one-factor model separately to each of the groups in turn. In this model the latent scale must be identified separately for each group, by constraining $\kappa^{(g)} = 0$ and $\phi^{(g)} = 1$ for every g. The separate measurement models are achieved by specifying the same model once under the overall `Model:` command and once under the group-specific commands `Model Bul:` and `Model Cyp:`, i.e. for all but the reference group `Bel`.

Figure 2: Mplus input syntax for a multigroup factor analysis model with 1 factor (Model N2).

```
Title: LCAT_FA_N2MG
    LCAT: examples of multiple-group latent variable models
    Linear factor analysis, one factor, Model N2
        Factor means and variances depend on country
        Measurement non-equivalence in one item
            Measurement variance and intercept non-equivalent
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-factor-analysis
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = tparl-tparties;
    Grouping = country(1=Bel 2=Bul 4=Cyp);
Model:
!! Basic measurement model:
    trust BY tparl* tlegal-tparties;  ! Not in E2B
!    trust BY tparl tlegal-tparties;  ! E2B
!    [tparl@0]; ! E2B
!! Error variances equal across countries:
!    tparl-tparties (3-7); ! In models E0,E1,E2,E2B,E3
!! Error variance of tparl varies across countries:
    tlegal-tparties (3-6); ! N1,N2,N3,N4,N5
!! Error correlation between two items:
!! Equal in all countries
!   tpolitic WITH tparties (8); ! E3,N4
!! Varies across countries
!   tpolitic WITH tparties; ! N5
!! Factor variance equal across countries:
!    trust@1 (1); ! E0,E1
!! Factor mean equal across countries:
!    [trust@0] (2); ! E0
!! Factor variance fixed at 1 in 1 country:
Model Bel:
    trust@1; ! Not in E2B
!    [trust*]; ! E2B
Model Bul: ! N2,N3,N4,N5
!! Item intercept of tparl varies across countries:
    [tparl]; ! N2, N3, N4, N5
!! Item loading of tparl varies across countries:
!    trust BY tparl; ! N3,N4,N5
Model Cyp: ! N2,N3,N4,N5
!! Item intercept of tparl varies across countries:
    [tparl]; ! N2, N3, N4, N5
!! Item loading of tparl varies across countries:
!    trust BY tparl; ! N3,N4,N5
```

Figure 3: Mplus input syntax for a multigroup factor analysis model which is equivalent to fitting a 1-factor model separately in each of the groups (Model N6).

```
Title: LCAT_FA_N6MG
    LCAT: examples of multiple-group latent variable models
    Linear factor analysis, one factor, Model N6
    Complete non-equivalence, i.e.
        separate one-factor models fitted to each group
     - see http://stats.lse.ac.uk/lcat/?resources=computing-factor-analysis
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = tparl-tparties;
    Grouping = country(1=Bel 2=Bul 4=Cyp);
Model:
!! Basic measurement model:
    trust BY tparl* tlegal-tparties;
!! Factor variance fixed at 1 in all countries:
    trust@1 (1);
!! Factor mean fixed at 0 in all countries:
    [trust@0] (2);
Model Bul:
!! Item intercepts vary across countries:
    [tparl-tparties];
!! Item loadings vary across countries
    trust BY tparl tlegal-tparties;
Model Cyp:
!! Item intercepts vary across countries:
    [tparl-tparties];
!! Item loadings vary across countries
    trust BY tparl tlegal-tparties;
```

3.3.2 Output

Consider now Mplus output for multigroup models with 1 factor. Its structure is always the same, so we show output only for one of the examples (model N2). This is given in Figure 4. Only part of the output is shown there. The omitted parts include other pieces of information, such as model selection statistics. These are included in Figure 5, which shows full output for the same model produced by the `lcat` post-processing functions in R (see Section 7).

The part of the output included in Figure 4 shows, for each model parameter, the point estimate of the parameter, its estimated standard error, their ratio — i.e. the Wald test statistic for the hypothesis that the parameter is 0 — and the P-value of this test against a two-sided alternative. If a parameter is fixed rather than estimable, the standard error is shown as 0.000.

The basic structure of this output table is that all parameters are shown for every group, even when a parameter is constrained to be equal across groups. In Figure 4, all the estimates are shown for the first group (`BEL`; group 1 below), but only selected ones for the other two (`BUL`, 2, and `CYP`, 3). Different types of parameters are labelled as follows:

- `Means`: Factor means $\kappa^{(g)}$. Here $\kappa^{(1)} = 0$ (constrained rather than estimated), $\hat{\kappa}^{(2)} = -1.302$ and $\hat{\kappa}^{(3)} = 0.205$.
- `Variances`: Factor variances $\phi^{(g)}$, $\phi^{(1)} = 1$ (constrained), $\hat{\phi}^{(2)} = 1.064$ and $\hat{\phi}^{(3)} = 1.261$.
- `Intercepts`: Intercepts $\tau_j^{(g)}$ of the measurement model. Here for `tparl` (item 1) these vary across groups, with $\hat{\tau}_1^{(1)} = 4.576$, $\hat{\tau}_1^{(2)} = 4.110$ and $\hat{\tau}_1^{(3)} = 5.082$. For item 2 (`tlegal`), for example, the intercepts do not vary across groups, and $\hat{\tau}_2^{(g)} = \hat{\tau}_2 = 5.029$.
- `<factor> BY <item>`: Loadings $\lambda_j^{(g)}$ of the measurement model of an item `<item>` as a measure of factor `<factor>`, here `TRUST`. Here none of the loadings vary across the groups, and for example $\hat{\lambda}_1^{(g)} = \hat{\lambda}_1 = 1.689$.
- `Residual Variances`: Error variances $\theta_j^{(g)}$ of the measurement model. Here none of these vary across the groups, and for example $\hat{\theta}_1^{(g)} = \hat{\theta}_1 = 2.071$.

The only type of parameter of a 1-factor model that is not included in this output is an error covariance $\theta_{jj'}^{(g)}$, as in models E3, N4 and N5. This would be listed in the following form:

```
TPOLITIC WITH TPARTIES       1.273     0.043     29.737     0.000
```

We mention already here new types of parameters that will appear only in multi-factor models. These parameters arise if the model includes a regression model where one factor (`ETA1`, say) is an explanatory variable for another factor (`ETA2`, say). The parameters of this model are listed in this output under the following headings:

- intercepts under `Intercepts`,
- regression coefficients under `ETA2 ON ETA1`, and
- residual variances under `Residual Variances`.

Figure 4: Part of Mplus output for a multigroup factor analysis model with 1 factor (N2).

```
MODEL RESULTS
                                                   Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value
Group BEL
 TRUST    BY
    TPARL              1.689      0.036     46.627      0.000
    TLEGAL             1.811      0.039     46.750      0.000
    TPOLICE            1.615      0.037     43.504      0.000
    TPOLITIC           1.881      0.035     53.392      0.000
    TPARTIES           1.807      0.034     52.652      0.000
 Means
    TRUST              0.000      0.000    999.000    999.000
 Intercepts
    TPARL              4.576      0.053     86.241      0.000
    TLEGAL             5.029      0.053     95.455      0.000
    TPOLICE            5.598      0.050    111.003      0.000
    TPOLITIC           4.059      0.047     85.769      0.000
    TPARTIES           4.015      0.046     86.711      0.000
 Variances
    TRUST              1.000      0.000    999.000    999.000
 Residual Variances
    TPARL              2.071      0.078     26.430      0.000
    TLEGAL             3.451      0.077     44.586      0.000
    TPOLICE            4.230      0.090     47.215      0.000
    TPOLITIC           0.540      0.022     24.513      0.000
    TPARTIES           0.825      0.024     34.814      0.000

Group BUL
 TRUST    BY
    TPARL              1.689      0.036     46.627      0.000
    TLEGAL             1.811      0.039     46.750      0.000
 Means
    TRUST             -1.302      0.042    -31.363      0.000
 Intercepts
    TPARL              4.110      0.060     68.827      0.000
    TLEGAL             5.029      0.053     95.455      0.000
 Variances
    TRUST              1.064      0.052     20.366      0.000
 Residual Variances
    TPARL              1.691      0.061     27.655      0.000
    TLEGAL             3.451      0.077     44.586      0.000

Group CYP
 TRUST    BY
    TPARL              1.689      0.036     46.627      0.000
    TLEGAL             1.811      0.039     46.750      0.000
 Means
    TRUST              0.205      0.042      4.906      0.000
 Intercepts
    TPARL              5.082      0.066     77.302      0.000
    TLEGAL             5.029      0.053     95.455      0.000
 Variances
    TRUST              1.261      0.071     17.683      0.000
 Residual Variances
    TPARL              2.708      0.122     22.135      0.000
    TLEGAL             3.451      0.077     44.586      0.000
```

Figure 5: Output from the `lcat` post-processing functions in R for a multigroup factor analysis model with 1 factor (Model N2) fitted in Mplus.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_fa_n2mg
Factor analysis model with 1 latent factor: TRUST

5 observed continuous items:
TPARL     TLEGAL    TPOLICE   TPOLITIC  TPARTIES
Multiple group model, with 3 groups:
BEL  BUL  CYP

Model estimates:
N = 5161        parameters = 23         log-likelihood = -49411.95
AIC = 98869.91  BIC = 99020.53
chi2-test = 2903.378    df = 37         P-value  <0.001
CFI = 0.834     TLI = 0.865     RMSEA = 0.212 [90% c.i.=( 0.206 - 0.219 )]

Models for the the latent factors:

Factor  TRUST :
      Mean    sd
BEL  0.000 1.000
BUL -1.302 1.032
CYP  0.205 1.123

Measurement parameters:
For items that are invariant across groups:

         Intercept TRUST resid.sd
TLEGAL       5.029 1.811    1.858
TPOLICE      5.598 1.615    2.057
TPOLITIC     4.059 1.881    0.735
TPARTIES     4.015 1.807    0.908

For items that are not invariant across groups:

TPARL :
    Intercept TRUST resid.sd
BEL     4.576 1.689    1.439
BUL     4.110 1.689    1.300
CYP     5.082 1.689    1.646
--------------------------------------------------------------------------------
```

3.4 2-factor multigroup models in Mplus

3.4.1 Input

Recall that in the 2-factor multigroup model, in the multiple-group formulation, the measurement model for item j in group g is

$$y_j^{(g)} = \tau_j^{(g)} + \lambda_{1j}^{(g)}\eta_1^{(g)} + \lambda_{2j}^{(g)}\eta_2^{(g)} + \epsilon_j^{(g)} \qquad \text{with } \epsilon_j \sim N(0, \theta_j^{(g)}), \tag{19}$$

for which we assume here that $\lambda_{1j}^{(g)} = 0$ or $\lambda_{2j}^{(g)} = 0$ for every j. The structural model for $\boldsymbol{\eta} = (\eta_1, \eta_2)$ can be specified in two equivalent ways, with the *covariance specification*

$$\eta_j^{(g)} \sim N(\kappa_j^{(g)}, \phi_{jj}^{(g)}) \qquad \text{for } j = 1, 2, \tag{20}$$

$$\text{cov}(\eta_1^{(g)}, \eta_2^{(g)}) = \phi_{12}^{(g)} \tag{21}$$

or with the *regression specification* such as

$$\eta_1^{(g)} \sim N(\kappa_1^{(g)}, \phi_{11}^{(g)}), \tag{22}$$

$$\eta_2^{(g)} = \gamma_0^{(g)} + \gamma_1^{(g)}\eta_1^{(g)} + \zeta^{(g)} \qquad \text{with } \zeta^{(g)} \sim N(0, \psi^{(g)}). \tag{23}$$

We will consider parallel examples of each of these, as summarised in Table 2. We now consider a model with two latent factors, *interpersonal trust* measured by three observed indicators (*ptrust*, *pfair* and *phelp*) and *institutional trust* measured by five variables (*tparl* to *tparties*). In the regression specification, personal trust will be treated as the explanatory variable in the structural model (η_1) and institutional trust as the response variable (η_2).

Mplus input for one model (E3) is shown, in Figure 6 for the covariance specification and in Figure 7 for the regression specification. Syntax for the other cases is again obtained by commenting and uncommenting lines from these, as indicated by comments in the syntax. The full syntax for the examples is also available at the LCAT website (`http://stats.lse.ac.uk/lcat/`).

Multigroup analysis is invoked as in the one-factor case, with the GROUPING option of the VARIABLE command, here `Grouping = country(1=Bel 2=Bul 4=Cyp);`.

The cases we discuss differ mainly in what constraints are imposed on the structural model, i.e. which parameters of this model do and do not vary across groups. The specification of the measurement model is essentially similar to the one-factor case. Most of the examples have complete measurement equivalence. The measurement model is then specified by the lines

```
Model:
!! Basic measurement model:
    perstrust BY ptrust* pfair phelp;
    insttrust BY tparl* tlegal-tparties;
!! Error variances equal across countries:
    ptrust-phelp (8-10);
    tlegal-tparties (4-7);
    tparl (3)
Model Bel:
    perstrust@1; insttrust@1;
    [perstrust@0]; [insttrust@0];
```

Table 2: Summary of the 2-factor models considered in Section 3.4. See equations (19)–(23) for the notation, and Figures 6 and 7 for full input syntax for Model E3 with the covariance and regression specifications respectively.

		Parameters that are constant across groups, in two different specifications:	
	Summary of structural model:	Covariance	Regression
E1	All parameters vary across groups		
E2	Factor variances do not vary	(ϕ_{11}, ϕ_{22})	(ϕ_{11}, ψ)
E3	(Conditional) variance of η_2 and factor association do not vary	(ϕ_{12}, ϕ_{22})	(γ_1, ψ)
E4	η_2 is marginally or conditionally independent of group	$(\phi_{12}, \phi_{22}, \kappa_2)$	$(\gamma_1, \psi, \gamma_0)$
E5	η_2 and η_1 are independent given group	$\phi_{12} = 0$	$\gamma_1 = 0$
N1	Like E1, but also non-equivalence of measurement in one item		

which also show how the means and variances of the two factors (labelled `perstrust` and `insttrust`) in one group (Belgium) are constrained for identifiability. Specification of non-equivalent measurement models is done in the same way as for one-factor models, as shown in one example. We consider the following models:

- **E1**: Unconstrained structural model, where all of its parameters vary across groups.
  - Covariance specification (E1C): Apart from the identifiability conditions on the latent scales (see above), nothing needs to be stated on the structural model. Covariance specification is then used by default, and the two factors are taken to be associated. All parameters of the structural model vary across groups by default.
  - Regression specification (E1R): This is invoked by

    ```
    Model:  insttrust ON perstrust;
    ```

    which specifies that `insttrust` is a response to `perstrust`. All of the parameters of this model, and of the distribution of `perstrust`, vary across groups by default.
- **E2**: Structural variance parameters are constant across groups, and constrained at 1.
  - Covariance specification (E2C): Since all structural parameters vary across groups by default, constraining them not to vary must always be stated explicitly. Here this is achieved by

    ```
    Model:  perstrust@1 (1); insttrust@1 (2);
    ```

    These specify that $\phi_{11}^{(g)} = \phi_{11} = 1$ and $\phi_{22}^{(g)} = \phi_{22} = 1$.
  - Regression specification (E2R): Same as E1R, plus the same added commands as in E2C. Here the line `insttrust@1 (2);` constrains the *residual* variance of the response variable `insttrust`, i.e. $\psi^{(g)} = \psi = 1$. This structural model is effectively a combination of two linear regression models with constant residual variances, one for η_1 (`perstrust`) given group and one for η_2 (`insttrust`) given group and η_1.

- **E3**: Variance parameter of `insttrust` and the association between the two factors are constant across groups.
  - Covariance specification (E3C): This is achieved with
    ```
    Model:
    insttrust@1 (2);
    perstrust WITH insttrust (13);
    ```
    These specify that $\phi_{22}^{(g)} = \phi_{22} = 1$ and $\phi_{12}^{(g)} = \phi_{12}$ respectively. Recall that the constraint is imposed by a number in the parentheses; what that number is does not matter, as long as it is not already used to identify some other paraneter constraint.
  - Regression specification (E3R): Commands
    ```
    Model:
    insttrust@1 (2);
    insttrust ON perstrust (13);
    ```
    These specify that $\psi^{(g)} = \psi = 1$ and $\gamma_1^{(g)} = \gamma_1$ respectively. Here the model for η_2 (`insttrust`) is a linear regression model where the residual variance is constant and the explanatory variables are the main effects of group and η_1 but not their interaction, so that the effect of η_1 on η_2 is the same in all groups.
- **E4**: Like E3, but in addition the mean (or intercept) parameter of `insttrust` is constant.
  - Covariance specification (E4C): Like E3C, plus
    ```
    Model:  [insttrust@0] (12);
    ```
    so that $\kappa_2^{(g)} = \kappa_2 = 0$. Here `insttrust` is marginally independent of group.
  - Regression specification (E4R): Like E3R, plus
    ```
    Model:  [insttrust@0] (12);
    ```
    which specifies that $\gamma_0^{(g)} = \gamma_0 = 0$. Here `insttrust` is conditionally independent of group given `perstrust`.
- **E5**: Like E1, except that the association parameter between the factors is 0 in all groups. This means that `perstrust` and `insttrust` are conditionally independent given group.
  - Covariance specification (E5C): Like E1C, plus
    ```
    Model:  perstrust WITH insttrust@0 (14);
    ```
    which specifies that $\phi_{12}^{(g)} = \phi_{12} = 0$.
  - Regression specification (E5R): Like E1R, except that the structural regression model is constrained by
    ```
    Model:  insttrust ON perstrust@0 (14);
    ```
    which specifies that $\gamma_1^{(g)} = \gamma_1 = 0$.
- **N1**: Like E1, but in addition all the measurement parameters of one item (`tparl`) depend on group. This is done in the same way for both specifications, and in the same way as for one-factor models. First, the error variance is allowed to vary, by omitting the constraint line `tparl (3);`. Second, the item intercept and loading are freed with group-specific `Model` commands, as shown by the commented-out command lines at the ends of the syntax files in Figures 6 and 7.

It should be noted that for models E1, E5 and N1 the two specifications are equivalent, i.e. specify the same model. In the other cases the covariance and regression specifications imply somewhat analogous but not identical models.

Figure 6: Mplus input syntax for a multigroup factor analysis model with 2 factors, in a covariance specification (Model E3C).

```
Title: LCAT_FA2_E3C
    LCAT: examples of multiple-group latent variable models
    Linear factor analysis, two factors, Model E3C
        Covariance specification
        Covariance between factors and variance of one factor do not depend on country
        Measurement equivalence in all items
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-factor-analysis
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = ptrust-phelp tparl-tparties;
    Grouping = country(1=Bel 2=Bul 4=Cyp);
Model:
!! Basic measurement model:
    perstrust BY ptrust* pfair phelp;
    insttrust BY tparl* tlegal-tparties;
!! Error variances equal across countries:
    ptrust-phelp (8-10);
    tlegal-tparties (4-7);
    tparl (3); ! All but model N1C
!! Factor variances equal across countries:
!    perstrust@1 (1); ! E2C
    insttrust@1 (2); ! E2C, E3C, E4C
!! Factor means equal across countries:
!    [insttrust@0] (12); ! E4C
!! Factor covariance equal across countries
    perstrust WITH insttrust (13); ! E3C, E4C
!! Factor covariance equal to 0 in all countries
!    perstrust WITH insttrust@0 (14); ! E5C
!! Factor variance fixed at 1 in 1 country:
Model Bel:
    perstrust@1;
    insttrust@1;
    [perstrust@0];
    [insttrust@0];
!Model Bul: ! N1C
!! Item intercept of tparl varies across countries:
!    [tparl]; ! N1C
!! Item loading of tparl varies across countries:
!    insttrust BY tparl; ! N1C
!Model Cyp: ! N1C
!! Item intercept of tparl varies across countries:
!    [tparl]; ! N1C
!! Item loading of tparl varies across countries:
!    insttrust BY tparl; ! N1C
```

Figure 7: Mplus input syntax for a multigroup factor analysis model with 2 factors, in a regression specification (Model E3R).

```
Title: LCAT_FA2_E3R
    LCAT: examples of multiple-group latent variable models
    Linear factor analysis, two factors, Model E3R
        Regression specification
        Regression coefficient between the factors and
            residual variance of second factor do not vary across countries
        Measurement equivalence in all items
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-factor-analysis
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = ptrust-phelp tparl-tparties;
    Grouping = country(1=Bel 2=Bul 4=Cyp);
Model:
!! Basic measurement model:
    perstrust BY ptrust* pfair phelp;
    insttrust BY tparl* tlegal-tparties;
!! Error variances equal across countries:
    ptrust-phelp (8-10);
    tlegal-tparties (4-7);
    tparl (3); ! All but model N1R
!! Regression model for the factors
!    insttrust ON perstrust; ! E1R, E2R, N1R
!! Factor association (regression coefficient) equal across countries
   insttrust ON perstrust (13); ! E3R, E4R
!! Factor covariance equal to 0 in all countries
!    insttrust ON perstrust@0 (14); ! E5R
!! Factor variances and residual variances equal across countries:
!    perstrust@1 (1); ! E2R
    insttrust@1 (2); ! E2R, E3R, E4R
!! Factor intercepts for one factor equal across countries:
!    [insttrust@0] (12); ! E4R
!! Factor variance fixed at 1 in 1 country:
Model Bel:
    perstrust@1;
    insttrust@1;
    [perstrust@0];
    [insttrust@0];
!Model Bul: ! N1R
!! Item intercept of tparl varies across countries:
!    [tparl]; ! N1R
!! Item loading of tparl varies across countries:
!    insttrust BY tparl; ! N1R
!Model Cyp: ! N1R
!! Item intercept of tparl varies across countries:
!    [tparl]; ! N1R
!! Item loading of tparl varies across countries:
!    insttrust BY tparl; ! N1R
```

3.4.2 Output

Structure of the Mplus output for 2-factor multigroup models is essentially the same as for 1-factor models. All parameter estimates are again listed for every group. Figure 8 shows part of the output for Model E3, from both the covariance (E3C) and regression (E3R) specifications. Only estimates of the parameters of the structural model are shown, and for two groups only (Belgium and Bulgaria). Parameters of measurement models are displayed exactly as for 1-factor models, so they are not shown here. Full output for these models, from the `lcat` post-processing functions in R (see Section 7) are shown in Figures 9 (for E3C) and 10 (for E3R).

Below we number the factors `perstrust` as η_1 and `insttrust` as η_2, and the two groups shown in the output as Belgium 1 and Bulgaria 2. For the covariance specification, different types of parameters of the structural model are labelled as follows:

- `Means`: Factor means $\kappa_j^{(g)}$, e.g. $\hat{\kappa}_1^{(2)} = -1.254$.
- `Variances`: Factor variances $\phi_{jj}^{(g)}$, e.g. $\hat{\phi}_{11}^{(2)} = 2.118$.
- `<factor> WITH <factor>`: Factor covariances $\phi_{12}^{(g)}$, here $\hat{\phi}_{12}^{(g)} = \hat{\phi}_{12} = 0.433$.

The same labelling of the factor means and variances is used in the regression formulation for any factor which is a not response variable to another factor, as `perstrust` (η_1) here. For a factor which is a response variable, the following labelling is used:

- `Intercepts`: Regression intercepts $\gamma_0^{(g)}$, e.g. $\hat{\gamma}_0^{(2)} = -1.052$.
- `<response_factor> ON <explanatory_factor>`: Regression coefficients $\gamma_1^{(g)}$, here $\hat{\gamma}_1^{(2)} = \hat{\gamma}_1 = 0.254$.
- `Residual Variances`: Residual variances $\psi^{(g)}$ in the structural regression model, here $\psi^{(g)} = \psi = 1$.

Finally, we note two features of the `lcat` function output, examples of which are shown in Figures 9 and 10. First, for the measurement model estimates of all loadings $\lambda_{jk}^{(g)}$ are shown, even where these are constrained to be 0. Second, to display the structural model the factors are ordered in such a way that the first to be shown is not a response to the other factor, and means and standard deviations of this factor across the groups are shown. For the second factor, the output shows either its marginal mean and standard deviation and covariance with the first factor (for the covariance specification) or intercept, regression coefficient and residual standard deviation of the model given the first factor (for the regression specification). The different cases are identified by labels of the parameters in the table. A similar convention is used for models with more than two factors, but such examples are not included in this document.

Figure 8: Part of Mplus output for a multigroup factor analysis model with 2 factors (Model E3). Output for both the covariance specification (E3C) and regression specification (E3R) are shown. Only estimates of the parameters of the structural model are shown, for two groups only.

```
------------------------------------------------------------
Covariance specification (Model E3C):
------------------------------------------------------------
MODEL RESULTS
                                                  Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value
Group BEL
 PERSTRUS WITH
    INSTTRUST          0.433      0.020     21.541      0.000
 Means
    PERSTRUST          0.000      0.000    999.000    999.000
    INSTTRUST          0.000      0.000    999.000    999.000
 Variances
    PERSTRUST          1.000      0.000    999.000    999.000
    INSTTRUST          1.000      0.000    999.000    999.000

Group BUL
 PERSTRUS WITH
    INSTTRUST          0.433      0.020     21.541      0.000
 Means
    PERSTRUST         -1.254      0.053    -23.683      0.000
    INSTTRUST         -1.291      0.036    -35.635      0.000
 Variances
    PERSTRUST          2.118      0.117     18.096      0.000
    INSTTRUST          1.000      0.000    999.000    999.000

------------------------------------------------------------
Regression specification (Model E3R):
------------------------------------------------------------
Group BEL
 INSTTRUS ON
    PERSTRUST          0.254      0.015     16.651      0.000
 Means
    PERSTRUST          0.000      0.000    999.000    999.000
 Intercepts
    INSTTRUST          0.000      0.000    999.000    999.000
 Variances
    PERSTRUST          1.000      0.000    999.000    999.000
 Residual Variances
    INSTTRUST          1.000      0.000    999.000    999.000

Group BUL
 INSTTRUS ON
    PERSTRUST          0.254      0.015     16.651      0.000
 Means
    PERSTRUST         -1.184      0.051    -23.155      0.000
 Intercepts
    INSTTRUST         -1.052      0.039    -27.056      0.000
 Variances
    PERSTRUST          1.889      0.112     16.833      0.000
 Residual Variances
    INSTTRUST          1.000      0.000    999.000    999.000
------------------------------------------------------------
```

Figure 9: Output from the `lcat` post-processing functions in R for a multigroup factor analysis model with 2 factors (with a covariance specification, Model E3C) fitted in Mplus.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_fa2_e3c
Factor analysis model with 2 latent factors: PERSTRUS INSTTRUS

8 observed continuous items:
PTRUST    PFAIR     PHELP     TPARL     TLEGAL    TPOLICE   TPOLITIC  TPARTIES
Multiple group model, with 3 groups:
BEL  BUL  CYP

Model estimates:
N = 5201        parameters = 31         log-likelihood = -82037.54
AIC = 164137.1  BIC = 164340.3
chi2-test = 3706.952    df = 101        P-value  <0.001
CFI = 0.839     TLI = 0.866     RMSEA = 0.144 [90% c.i.=( 0.14 - 0.147 )]

Models for the the latent factors:

Factor  PERSTRUS :
      Mean    sd
BEL  0.000 1.000
BUL -1.254 1.455
CYP -0.529 1.460

Factor  INSTTRUS :
      Mean cov.PERSTRUS sd
BEL  0.000        0.433  1
BUL -1.291        0.433  1
CYP  0.236        0.433  1

Measurement parameters:
For items that are invariant across groups:

         Intercept PERSTRUS INSTTRUS resid.sd
PTRUST       5.223    1.439    0.000    1.548
PFAIR        5.838    1.322    0.000    1.421
PHELP        4.750    1.166    0.000    1.656
TPARL        4.578    0.000    1.893    1.442
TLEGAL       5.047    0.000    1.887    1.819
TPOLICE      5.612    0.000    1.677    2.036
TPOLITIC     4.058    0.000    1.921    0.782
TPARTIES     4.015    0.000    1.847    0.935
--------------------------------------------------------------------------------
```

Figure 10: Output from the `lcat` post-processing functions in R for a multigroup factor analysis model with 2 factors (with a regression specification, Model E3C) fitted in Mplus.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_fa2_e3r
Factor analysis model with 2 latent factors: PERSTRUS INSTTRUS

8 observed continuous items:
PTRUST    PFAIR     PHELP     TPARL     TLEGAL    TPOLICE   TPOLITIC  TPARTIES
Multiple group model, with 3 groups:
BEL  BUL  CYP

Model estimates:
N = 5201        parameters = 31         log-likelihood = -82083.42
AIC = 164228.8  BIC = 164432.1
chi2-test = 3798.7      df = 101        P-value  <0.001
CFI = 0.835     TLI = 0.863     RMSEA = 0.145 [90% c.i.=( 0.141 - 0.149 )]

Models for the the latent factors:

Factor  PERSTRUS :
      Mean    sd
BEL  0.000 1.000
BUL -1.184 1.374
CYP -0.503 1.292

Factor  INSTTRUS :
    Intercept b.PERSTRUS resid.sd
BEL     0.000      0.254        1
BUL    -1.052      0.254        1
CYP     0.374      0.254        1

Measurement parameters:
For items that are invariant across groups:

         Intercept PERSTRUS INSTTRUS resid.sd
PTRUST       5.220    1.518    0.000    1.555
PFAIR        5.843    1.406    0.000    1.406
PHELP        4.750    1.231    0.000    1.658
TPARL        4.577    0.000    1.806    1.443
TLEGAL       5.046    0.000    1.800    1.820
TPOLICE      5.611    0.000    1.599    2.038
TPOLITIC     4.058    0.000    1.833    0.779
TPARTIES     4.015    0.000    1.764    0.933
--------------------------------------------------------------------------------
```

4 Latent trait models for multiple groups

4.1 General specification of the models

By a multigroup *latent trait model*, we mean a model of the general form (4) where the latent variables (*latent traits*) $\boldsymbol{\eta}$ are continuous but the observed items $\mathbf{y}$ are categorical variables. Latent trait models are also often called Item Response Theory (IRT) models.

The specification of the structural model for the latent traits is exactly similar as for the latent factors in a linear factor analysis model. We again consider only the cases of one and two traits, but generalisation to models with more traits is straightforward. Repeating the expressions from Section 3.1, in the one-trait case we assume that the structural model for the latent trait η in group $g = 1, \dots, G$ is

$$\eta^{(g)} \sim N(\kappa^{(g)}, \phi^{(g)}). \tag{24}$$

In the two-trait case the model for the traits $\boldsymbol{\eta} = (\eta_1, \eta_2)$ in group g is given by the covariance specification

$$\begin{aligned} \eta_j^{(g)} &\sim N(\kappa_j^{(g)}, \phi_{jj}^{(g)}) \qquad \text{for } j = 1, 2, & (25)\\ \text{cov}(\eta_1^{(g)}, \eta_2^{(g)}) &= \phi_{12}^{(g)} & (26) \end{aligned}$$

or equivalently with a regression specification such as

$$\begin{aligned} \eta_1^{(g)} &\sim N(\kappa_1^{(g)}, \phi_{11}^{(g)}), & (27)\\ \eta_2^{(g)} &= \gamma_0^{(g)} + \gamma_1^{(g)}\eta_1^{(g)} + \zeta^{(g)} \qquad \text{with } \zeta^{(g)} \sim N(0, \psi^{(g)}). & (28) \end{aligned}$$

To identify the scales of the latent traits, we assume throughout that $(\kappa^{(G)}, \phi^{(G)}) = (0, 1)$ for one trait, and $(\kappa_1^{(G)}, \phi_{11}^{(G)}, \kappa_{22}^{(G)}, \phi_{22}^{(G)}) = (0, 1, 0, 1)$ or $(\kappa_1^{(G)}, \phi_{11}^{(G)}, \gamma_0^{(G)}, \psi^{(G)}) = (0, 1, 0, 1)$ for two traits. Here we use the highest rather than lowest-numbered group as the reference group, because that turns out to be convenient for one way of fitting the models in Mplus.

In this section we we will discuss separately *multiple-group specifications* and *covariate specifications* of models, because these correspond also to two distinct ways of specifying the models in Mplus (see Section 2 for a conceptual introduction to this notational distinction). The formulas (24)–(28) represent a multiple-group formulation of the structural model for latent traits. For a covariate specification, define the covariate vector $\mathbf{x}^{(g)} = (x_1^{(g)}, \dots, x_{G-1}^{(g)})$ of dummy variables, where for an observation from group g we have $x_g^{(g)} = 1$ and $x_{g'}^{(g)} = 0$ for all $g' \neq g$. The covariate formulations of structural models we will consider are

$$\eta^{(g)} \sim N(\kappa + \kappa^{(1)}x_1^{(g)} + \dots + \kappa^{(G-1)}x_{G-1}^{(g)}, \phi) \tag{29}$$

in the one-trait case, and

$$\begin{aligned} \eta_j^{(g)} &\sim N(\kappa_j + \kappa_j^{(1)}x_1^{(g)} + \dots + \kappa_j^{(G-1)}x_{G-1}^{(g)}, \phi_{jj}) \qquad \text{for } j = 1, 2, & (30)\\ \text{cov}(\eta_1^{(g)}, \eta_2^{(g)}) &= \phi_{12} + \phi_{12}^{(1)}x_1^{(g)} + \dots + \phi_{12}^{(G-1)}x_1^{(G-1)} & (31) \end{aligned}$$

or

$$\begin{aligned} \eta_1^{(g)} &\sim N(\kappa_1 + \kappa_1^{(1)}x_1^{(g)} + \dots + \kappa_1^{(G-1)}x_{G-1}^{(g)}, \phi_{11}), & (32)\\ \eta_2^{(g)} &= [\gamma_0 + \gamma_0^{(1)}x_1^{(g)} + \dots + \gamma_0^{(G-1)}x_{G-1}^{(g)}] \\ &\quad + [\gamma_1 + \gamma_1^{(1)}x_1^{(g)} + \dots + \gamma_1^{(G-1)}x_{G-1}^{(g)}]\,\eta_1^{(g)} + \zeta^{(g)} \qquad \text{with } \zeta^{(g)} \sim N(0, \psi) & (33) \end{aligned}$$

in the two-trait case. Note first that these differ from the multiple-group formulation (24)–(28) in that in (29)-(33) we assume that the variance parameters ϕ, ϕ_{11}, ϕ_{22} and ψ are constant across groups g. We make this simplification because a model without this assumption is not possible in the Mplus covariate specification (at least without fairly elaborate tricks). Even under this assumption, parameters with the same symbols are not always identical in our two sets of formulas, but they are analogous. For example, $\kappa^{(G)}$ and $\kappa^{(g)}$, $g < G$, in (24) correspond to κ and $\kappa + \kappa^{(g)}$ respectively in (29).

Consider now the observed items y_j, $j = 1, \ldots, p$. Here different items are always assumed to be conditionally independent of each other given the latent traits $\boldsymbol{\eta}$, so we can without loss of generality describe the measurement models for one item at a time. An item y_j is now assumed to be categorical, with L possible levels (*categories*) $l = 1, \ldots, L$ (different items can have different numbers of levels, so we should write L_j; we omit the subscript to simplify the notation). The item may be either *ordinal*, when the levels are taken to be ordered from 1 to L, or *nominal* when they are not ordered. When $L = 2$, an item is *binary* (dichotomous), which can be equivalently treated as either ordinal or nominal. In both ordinal and nominal cases we refer to levels 1 and L as the "lowest" and "highest" levels of an item respectively.

Consider the measurement model of an item y_j in group g given latent trait η in the one-trait case. The form of this model depends on whether the item is nominal or ordinal. For a nominal item, we use the *multinomial logistic model*

$$\pi_{jl}^{(g)} = P(y_j^{(g)} = l|\eta^{(g)}) = \frac{\exp(\tau_{jl}^{(g)} + \lambda_{jl}^{(g)}\eta^{(g)})}{\sum_{l'=1}^{L}\exp(\tau_{jl'}^{(g)} + \lambda_{jl'}^{(g)}\eta^{(g)})} \tag{34}$$

for $l = 1, \ldots, L$ and $g = 1, \ldots, G$. Mplus uses the identification condition that $\tau_{jL}^{(g)} = \lambda_{jL}^{(g)} = 0$ for all g, i.e. that the *highest* category of an item is the baseline category in this measurement model.

Equation (34) gives the multiple-group formulation of the multinomial logistic model. A covariate formulation of it is

$$\begin{aligned} \pi_{jl}^{(g)} &= P(y_j = l|\eta^{(g)}, \boldsymbol{x}^{(g)}) \qquad (35)\\ &= \frac{\exp\left([\tau_{jl} + \tau_{jl}^{(1)}x_1^{(g)} + \cdots + \tau_{jl}^{(G-1)}x_{G-1}^{(g)}] + [\lambda_{jl} + \lambda_{jl}^{(1)}x_1^{(g)} + \cdots + \lambda_{jl}^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}{\sum_{l'=1}^{L}\exp\left([\tau_{jl'} + \tau_{jl'}^{(1)}x_1^{(g)} + \cdots + \tau_{jl'}^{(G-1)}x_{G-1}^{(g)}] + [\lambda_{jl'} + \lambda_{jl'}^{(1)}x_1^{(g)} + \cdots + \lambda_{jl'}^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}. \end{aligned}$$

The two expressions are equivalent when we equate $\tau_l^{(G)}$ and $\lambda_l^{(G)}$ in (34) with τ_l and λ_l in (35) respectively, and $\tau_l^{(g)}$ and $\lambda_l^{(g)}$ for $g = 1, \ldots, G-1$ with $\tau_l + \tau_l^{(g)}$ and $\lambda_l + \lambda_l^{(g)}$ respectively.

For an ordinal item, we use the *ordinal logistic* model (proportional odds model), which in Mplus is parametrised in the multiple-group specification as

$$\nu_{jl}^{(g)} = P(y_j^{(g)} \leq l|\eta^{(g)}) = \frac{\exp(\tau_{jl}^{(g)} - \lambda_j^{(g)}\eta^{(g)})}{1 + \exp(\tau_{jl}^{(g)} - \lambda_j^{(g)}\eta^{(g)})} \tag{36}$$

for $l = 1, \ldots, L-1$. The probabilities of individual levels of y_j are given by $\pi_{jl}^{(g)} = \nu_{jl}^{(g)} - \nu_{j,l-1}^{(g)}$ for $l = 1, \ldots, L$, where we take $\nu_{j0} = 0$ and $\nu_{jL} = 1$. The direct equivalent of (37) in the covariate specification is

$$\begin{aligned}\nu_{jl}^{(g)} &= P(y_j \le l|\eta^{(g)}, \mathbf{x}^{(g)}) \qquad (37)\\ &= \frac{\exp\left([\tau_{jl}+\tau_{jl}^{(1)}x_1^{(g)}+\cdots+\tau_{jl}^{(G-1)}x_{G-1}^{(g)}]-[\lambda_j+\lambda_j^{(1)}x_1^{(g)}+\cdots+\lambda_j^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}{1+\exp\left([\tau_{jl}+\tau_{jl}^{(1)}x_1^{(g)}+\cdots+\tau_{jl}^{(G-1)}x_{G-1}^{(g)}]-[\lambda_j+\lambda_j^{(1)}x_1^{(g)}+\cdots+\lambda_j^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}.\end{aligned}$$

However, when $L > 2$, it is not actually possible in Mplus (without tricks at least) to use an explicit covariate specification to implement (37) but only the more constrained model

$$\nu_{jl}^{(g)} = \frac{\exp\left([\tau_{jl}+\tau_j^{(1)}x_1^{(g)}+\cdots+\tau_j^{(G-1)}x_{G-1}^{(g)}]-[\lambda_j+\lambda_j^{(1)}x_1^{(g)}+\cdots+\lambda_j^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}{1+\exp\left([\tau_{jl}+\tau_j^{(1)}x_1^{(g)}+\cdots+\tau_j^{(G-1)}x_{G-1}^{(g)}]-[\lambda_j+\lambda_j^{(1)}x_1^{(g)}+\cdots+\lambda_j^{(G-1)}x_{G-1}^{(g)}]\,\eta^{(g)}\right)}, \qquad (38)$$

i.e. a model where, when comparing one group to another, the intercept terms for all the categories l are shifted by the same distance.

It is conventional to parametrise the ordinal logistic model as in (36) and (37), with negative signs for the loadings $\lambda_j^{(g)}$.[3] This means that for a binary item, for which the multinomial and ordinal models are equivalent, their estimated loadings will be opposites of each other. For a binary item the multinomial model (34) gives

$$\pi_{j1}^{(g)} = P(y_j^{(g)} = 1|\eta^{(g)}) = \frac{\exp(\tau_{j1}^{(g)} + \lambda_{j1}^{(g)}\eta^{(g)})}{1+\exp(\tau_{j1}^{(g)} + \lambda_{j1}^{(g)}\eta^{(g)})} \qquad (39)$$

and the ordinal model (36) gives

$$\nu_{j1}^{(g)} = \pi_{j1}^{(g)} = P(y_j^{(g)} = 1|\eta^{(g)}) = \frac{\exp(\tau_{j1}^{(g)} - \lambda_{j}^{(g)}\eta^{(g)})}{1+\exp(\tau_{j1}^{(g)} - \lambda_{j}^{(g)}\eta^{(g)})} \qquad (40)$$

which are equal when $\lambda_{j1}^{(g)} = -\lambda_j^{(g)}$; naturally $\pi_{j2}^{(g)} = 1 - \pi_{j1}^{(g)}$.

Measurement equivalence or lack of it is again determined by whether or not parameters of the measurement model are the same across groups. A nominal item y_j has measurement equivalence if in the multiple-group formulation (34) we have $\tau_{jl}^{(1)} = \cdots = \tau_{jl}^{(G)} = \tau_{jl}$ and $\lambda_{jl}^{(1)} = \cdots = \lambda_{jl}^{(G)} = \lambda_{jl}$ for all l, and an ordinal item if in (36) we have $\tau_{jl}^{(1)} = \cdots = \tau_{jl}^{(G)} = \tau_{jl}$ and $\lambda_j^{(1)} = \cdots = \lambda_j^{(G)} = \lambda_j$ for all l. In the covariate formulation, measurement equivalence holds if, for all l, $\tau_{jl}^{(1)} = \cdots = \tau_{jl}^{(G-1)} = 0$ and $\lambda_{jl}^{(1)} = \cdots = \lambda_{jl}^{(G-1)} = 0$ in (35) or $\tau_{jl}^{(1)} = \cdots = \tau_{jl}^{(G-1)} = 0$ and $\lambda_j^{(1)} = \cdots = \lambda_j^{(G-1)} = 0$ in (37). The latter form makes it clear that measurement equivalence holds when in the covariate formulation we have zero coefficients for all the terms in the measurement model which involve the group dummies $\mathbf{x}^{(g)} = (x_1^{(g)}, \ldots, x_{G-1}^{(g)})$.

Measurement models in the two-trait case are a straightforward generalisation of the ones above, and we omit their formulas. In all of our two-trait examples, the measurement model of each item has non-zero loadings for only one of the latent traits.

[3] However, in the output of the `lcat` functions in R (see Section 7) we reverse their signs by multiplying all the estimated loadings for the ordinal model by -1.

4.2 1-trait multigroup models in Mplus

4.2.1 Input

We use again the subset of data from the European Social Survey that is described in Appendix A. It involves respondents from three groups (countries), Belgium, Bulgaria and Cyprus. Example syntax for one 1-trait model is given in Figures 11 and 12, separately for a covariate specification and a multiple-group specification. Commands for the other examples are obtained by commenting in and out lines from this syntax, as discussed below. The syntax for all the examples is also available at the LCAT website (`http://stats.lse.ac.uk/lcat/`).

We discuss one-trait models for 3 of the observed items, *polinter* (4 categories), *polhard* (2 categories) and *polmind* (5 categories). Here the variable *polhard* is first dichotomised from its original 5-category version, just to show how this is done. This combines the levels 1–2 (new category 1) and 3–5 (2) of the original variable. The dichotomisation can be done within Mplus by using the CUT option of the DEFINE command, here

```
Define:  cut polhard(2);
```

The items are regarded as indicators of a single continuous latent trait, which we label *political engagement* (or political interest). Low values of *polinter* and *polhard*, but high values of *polmind*, correspond to high levels of engagement.

Given the wording of their response options, all of the items could be taken to be ordinal. However, to demonstrate all the possibilities, we will model *polmind* as nominal and *polinter* as ordinal ("categorical" in Mplus terminology). Since *polhard* is dichotomous, it could be declared equivalently as either nominal or ordinal, and we specify it as ordinal. With these choices, the specification of the variable types is throughout

```
Variable:
   Categorical = polmind polhard;
   Nominal = polinter;
```

This specification invokes the multinomial logistic measurement model (34 or 35) for *polinter*, and the ordinal logistic model (36 or 37) for *polmind* and *polhard*.

We consider five models, some with further subvariants, to illustrate various types of parameter constraints. These cases are summarised in Table 3. The labelling of the examples matches roughly that of some of the examples of 1-factor models in Section 3.3 (c.f. Table 1).

Mplus makes it possible to fit the multigroup latent trait model in two quite different ways, which implement the multiple-group and covariate specifications discussed above. We discuss the covariate specification first. Example commands for it are shown in Figure 11.

The covariate specification makes explicit use of dummy variables $\mathbf{x}^{(g)} = (x_1^{(g)}, \ldots, x_{G-1}^{(g)})$ for the groups. Here we leave the highest-numbered group as the reference group without a dummy variable. Any group can of course be used as the reference here, but this choice matches the multiple-group specification discussed below, where the highest-numbered group is the reference

Table 3: Summary of the 1-trait models considered in Section 4.2. See equations (24), (34) and (36) for the notation, and Figures 11 and 12 for full input syntax for Model N3C.

	Key features of the model:
Models with measurement equivalence across groups in all items	
E0	Trait means and variances equal across groups ($\kappa^{(g)} = 0$, $\theta^{(g)} = 1$ for all g)
E1	Trait variances equal across groups ($\theta^{(g)} = 1$ for all g), $\kappa^{(g)}$ varies
E2	Both trait means $\kappa^{(g)}$ and variances $\theta^{(g)}$ vary across groups g
Models with non-equivalence of measurement in one item j (A=polmind, B=polhard, C=polinter). In all of these, $\kappa^{(g)}$ and $\theta^{(g)}$ vary across groups.	
N2	Intercepts $\tau_{jl}^{(g)}$ vary across groups g, loadings $\lambda_{jl}^{(g)} = \lambda_{jl}$ and $\lambda_j^{(g)} = \lambda_j$ do not
N3	Both intercepts ($\tau_{jl}^{(g)}$) and loadings ($\lambda_{jl}^{(g)}$ or $\lambda_j^{(g)}$) vary across groups g
N6	Non-equivalence of measurement for all items, i.e. a 1-trait model fitted separately for each group

group by default. In our example this is Cyprus, so we need dummy variables for Belgium and Bulgaria. These can be generated in the DEFINE command, as follows:

```
Define:  bel = (country==1); bul = (country==2);
```

We will use maximum likelihood (ML) estimation in all latent trait models. This is requested with the ANALYSIS command, as

```
Analysis:  Estimator=ML;
```

There are also many other options for ANALYSIS, which for example adjust the settings of the iterative estimation algorithm. One of them that should always be used is STARTS, which is used to request multiple starting points for the algorithm. This and other issues in the numerical implementation of the estimation are discussed very briefly in Section 6.

In both the covariate and multiple-group specifications, the basic measurement model in our example is requested under the MODEL command with the lines

```
engage BY polhard* polmind;
polinter ON engage;
engage@1;
```

These lines declare the latent trait `engage`, which is measured by items `polhard`, `polmind` and `polinter`. The variance of the latent trait is fixed (in one or all groups, depending on the specification) at 1, and loadings of all the items are estimable parameters. This specification may then be modified by further commands to achieve particular cross-group differences or equalities, as discussed in the examples.

The format of this model specification is the same as for the linear factor analysis models discussed in Section 3. The difference arises because here the items have been declared `categorical` or `nominal`, from which Mplus knows to apply to them the corresponding multinomial and ordinal logistic, rather than linear, measurement models.

In the commands above we specified the measurement model for one item `polinter` with an ON rather than a BY command. This construction works (both here and for factor analysis) because the latent variable `engage` has previously been declared by a BY command, so the subsequent line `polinter ON engage` is meaningful. Here we use this approach because it shortens the code a little. For a nominal item like `polinter`, a corresponding ON command must refer to all of the intercept terms of the variable separately, as in

```
engage BY polinter#1 polinter#2 polinter#3;
[polinter#1](3); [polinter#2](4); [polinter#3](5);
```

The second line of this specifies that the intercept terms should be constant across the groups, to overrule the default that they vary across groups.

Consider now the specific commands for the covariate specification for the cases in Table 3. Unless otherwise mentioned, all of these lines come under the MODEL command (see Figure 11 for the full syntax). First, three models where full measurement equivalence holds:

- **E0**: The trait means and variances $\phi = 1$ are equal in all groups.
  `engage ON bel@0 bul@0;`
  This sets $\kappa^{(1)} = \kappa^{(2)} = 0$ in (29).
- **E1**: The trait means vary across groups, but the variances do not.
  `engage ON bel bul;`
  This lets $\kappa^{(1)}$ and $\kappa^{(2)}$ in (29) be estimated freely.
- **E2**: Both trait means and variances vary across groups. This is not possible with the standard covariate specification, i.e. it is not possible for ϕ in (29) to vary across groups.

Next, two models where some or all of the measurement parameters for one item may be different across groups g. For each, we show three versions, taking each of the three items in turn to be the one with non-equivalent measurement. The labelling of models in the comments in the code refers to these by letter, with A for *polmind* (ordinal), B for *polhard* (binary, fitted as ordinal) and C for *polinter* (nominal). In each case, all other measurement parameters and the trait variance $\phi = 1$ are equal across groups, and trait means $\kappa^{(g)}$ vary across groups.

- **N2**: Measurement intercepts vary across groups.
  - N2B: `polhard ON bel bul;` This fits model (37) with $L = 2$, with $\lambda_j^{(1)} = \cdots = \lambda_j^{(G-1)} = 0$ and non-zero $\tau_{jl}^{(g)}$. This is also a covariate-specification version of the binary logistic model (40).
  - N2C: `polinter ON bel bul;` This fits model (35), with $\lambda_{jl}^{(1)} = \ldots \lambda_{jl}^{(G-1)} = 0$ for all l, but non-zero $\tau_{jl}^{(g)}$.
  - N2A2: `polmind ON bel bul;` Because this variable is ordinal with more than 2 categories, this specification actually gives model (38) with $\lambda_j^{(1)} = \cdots = \lambda_j^{(G-1)} = 0$; this is labelled in the code Model N2A2, to distinguish it from N2A (equation 37) which can only be fitted using a multiple-group specification.

- **N3**: Both measurement intercepts and loadings vary across groups. These are models where the $\lambda_{jl}^{(g)}$ in (35) or $\lambda_j^{(g)}$ in (37) or (38) are non-zero. It can be seen that this leads to models involving interactions between group and latent trait, which appear in the model as the products $x_{g'}^{(g)}\eta^{(g)}$ of the trait and the group dummies. This is achieved in Mplus by creating these products explicitly, with the XWITH option:

```
bel_en | bel XWITH engage;
bul_en | bul XWITH engage;
```

  and then using these product variables (`bel_en` and `bul_en`) also in the model specification. This also requires the ANALYSIS option

```
Analysis:  Type = Random;
```

  - N3B: `polhard ON bel bul bel_en bul_en;` This fits model (37) with $L = 2$, with both $\lambda_j^{(g)}$ and $\tau_{jl}^{(g)}$ non-zero.
  - N3C: `polinter ON bel bul bel_en bul_en;` This fits model (35), with both $\lambda_{jl}^{(g)}$ and $\tau_{jl}^{(g)}$ non-zero.
  - N3A2: `polmind ON bel bul bel_en bul_en;` This fits model (38), with both $\lambda_j^{(g)}$ and $\tau_j^{(g)}$ non-zero.

Table 3 also mentions model N6, which involves fitting the one-trait model entirely separately for each of the groups. This model is conveniently possible in Mplus only with the multiple-group specification, which is discussed below.

The form of the multiple-group specification of the multigroup latent trait model is rather different from the covariate specification. It specifies a model with the syntax of a latent class model (c.f. Section 5), with the group variable as a "known-class" variable whose observed levels are equated with the latent classes with certainty. This is declared in the VARIABLE command by

```
Variable:
    Classes = countryx(3);
    Knownclass = countryx (country=1 country=2 country=4);
```

The known-class variable will then be called `countryx` and will have 3 levels, as many as there are groups. Its classes are equal to specific levels of the observed variable `country`, as stated in the `Knownclass` statement. The levels of the known-class variable are numbered in the order they are mentioned in parentheses in this statement, and the last of them (here country with value 4, i.e. Cyprus) will be used as the reference level in the structural model.

The multiple-group formulation requires the following ANALYSIS command:

```
Analysis:
    Type = Mixture;
    Estimator=ML;
    Algorithm = Integration;
```

and the following specification of the basic model in our examples:

```
Model:
    %overall%
        engage BY polhard* polmind;
        polinter ON engage;
        engage@1;
        [polinter#1](3); [polinter#2](4); [polinter#3](5);
```

The last line in this specifies that the intercept terms in the measurement model of the nominal item `polinter` should be ocnstant across the groups, to overrule the default where they vary across groups. This command is omitted in models where these intercepts should vary (such as examples N2C, N3C and N6 below).

The multiple-group (known-class) fits of the cases in Table 3 are then achieved as follows. Unless otherwise mentioned, all of these lines come under the MODEL command (see Figure 12 for the full syntax). Consider first the three cases with complete measurement equivalence:

- **E0**: The trait means and variances are equal in all groups.
  ```
  %overall%
      [engage@0] (2);
  ```
  This constrains $\kappa^{(g)} = \kappa = 0$ for all $g = 1, \ldots, G$ in (24). The variances $\phi^{(g)} = \phi = 1$ are equal by default.
- **E1**: The trait means vary across groups, but the variances do not. This is implied by the lines under the basic `%overall%` model statement above, so no further commands are needed.
- **E2**: Both trait means and variances vary across groups. We now need to free $\phi^{(g)}$ in the non-reference groups $g = 1, 2$. This is achieved by adding model statements that are specific to those groups, i.e. for corresponding levels of the known-class variable `countryx`:
  ```
  %countryx#1%
      engage;
  %countryx#2%
      engage;
  ```

Next, two models where some or all of the measurement parameters for one item may be different across groups g. For each, we show three versions, taking each of the three items in turn to be the one with non-equivalent measurement. The labelling of models in the comments in the code referes to these by letter, with A for *polmind* (ordinal), B for *polhard* (binary, fitted as ordinal) and C for *polinter* (nominal). In each case, all other measurement parameters are equal across groups, and trait means $\kappa^{(g)}$ vary across groups. The trait variance $\phi^{(g)} = \phi = 1$ are also equal across groups, to match the cases considered under the covariate specification. Here, however, the variances can also be freed, in the same way as in model E2.

Each of these models involves commands under the group-specific model statements `%country#1%` and `%country#2%`. In most cases these two are identical, so we list only one of them.

- **N2**: Measurement intercepts vary across groups.
  - N2B:
    ```
    %countryx#1%
        [polhard$1];
    ```
    This fits model (36) with $L = 2$, i.e. (40) with $\lambda_j^{(g)} = \lambda_j$ but $\tau_{j1}^{(g)}$ freely estimated.
  - N2C:
    ```
    %countryx#1%
      [polinter#1 polinter#2 polinter#3];
    ```
    This fits model (34), with $\lambda_{jl}^{(g)} = \lambda_{jl}$ for all l, but $\tau_{jl}^{(g)}$ freely estimated.
  - N2A:
    ```
    %countryx#1%
      [polmind$1 polmind$2 polmind$3 polmind$4];
    ```
    This fits model (36), with $\lambda_j^{(g)} = \lambda_j$ for all l, but $\tau_{jl}^{(g)}$ freely estimated.
  - N2A2: In the covariate specification, where N2A cannot be fitted, this was model (38) with $\lambda_j^{(1)} = \cdots = \lambda_j^{(G-1)} = 0$. The multiple-group equivalent of this is (36) with the additional constraints that the differences $\tau_{jl}^{(g)} - \tau_{jl}^{(G)} = \tau_j^{(g)}$ are constant for all levels l in each group g. This can be achieved in the multiple-group specification with the use of the MODEL CONSTRAINT command, as follows:
    ```
    Model:
    %overall%
       [polmind$1](a1); [polmind$2](a2); [polmind$3](a3); [polmind$4](a4);
    %countryx#1%
       [polmind$1](b1); [polmind$2](b2); [polmind$3](b3); [polmind$4](b4);
    %countryx#2%
       [polmind$1](c1); [polmind$2](c2); [polmind$3](c3); [polmind$4](c4);
    Model constraint:
        0=b1-a1+a2-b2; 0=b2-a2+a3-b3; 0=b3-a3+a4-b4;
        0=c1-a1+a2-c2; 0=c2-a2+a3-c3; 0=c3-a3+a4-c4;
    ```
    What happens here is that the intercept terms $\tau_{jl}^{(g)}$, i.e. `[polmind$1]` and so on, are assigned labels (`a1` to `c4`) under the model specifications, and the constraints are then stated in terms of these labels under the MODEL CONSTRAINT command. For example, the `0=b1-a1+a2-b2;` corresponds to the constraint $\tau_j^{(1)} = \tau_{j1}^{(1)} - \tau_{j1}^{(G)} = \tau_{j2}^{(1)} - \tau_{j2}^{(G)}$, i.e. $0 = \tau_{j1}^{(1)} - \tau_{j1}^{(G)} + \tau_{j2}^{(G)} - \tau_{j2}^{(1)}$ (with $G = 3$). These expressions (and those for N3A2 below) are given here mainly for completeness, and to illustrate the use of the model constraint command in Mplus. A measurement model like N2A2 does not possess any obvious advantage that would recommend its use in the multiple-group formulation in preference to the simpler N2A.

- **N3**: Both measurement intercepts and loadings, i.e. $\tau_{jl}^{(g)}$ and $\lambda_{jl}^{(g)}$ or $\lambda_j^{(g)}$ in (34) or (36), vary across groups g. This is achieved by taking the code for the N2 models (keeping the group-specific intercept statements) and adding group-specific commands for the loadings.
  - N3B:
    ```
    %countryx#1%
      engage BY polhard;
    ```

– N3C:

```
%countryx#1%
  polinter ON engage;
```

(note that this needs to match the form of the overall model statement for `polinter`, which was also given in the ON form)

– N3A:

```
%countryx#1%
  engage BY polmind;
```

– N3A2: This is model N3A, but with the added constraints on $\tau_{jl}^{(g)}$ imposed by N2A2. The code is like for N3A, but the specification of the intercept terms for `polmind` as in N2A2.

Consider finally model **N6**, which involves complete non-equivalence of measurement in all items, i.e. fitting a one-trait model separately in each group. This is achieved by using group-specific model statements for intercepts and loadings of all items, i.e. combining the measurement models of N3A, N3B and N3C. Because there are then no measurement models that link the groups, the distribution of the latent trait needs to be constrained as $(\kappa^{(g)}, \phi^{(g)}) = (0, 1)$ in all the groups. The structural model is thus specified as in model E0.

There are thus two different ways of specifying a multigroup latent trait model in Mplus, the covariate specification and the multiple-group specification. When both are possible for a model, they should produce equivalent estimated models. In practice, however, the multiple-group specification is typically preferable, for two main reasons. First, it can be used to fit also some models for which the covariate specification is not possible (or possible only with contrived computational tricks). Second, for some models the estimation is much faster with the multiple-group specification. The covariate specification is thus of limited interest for the latent trait models considered here. It is, however, something to keep in mind as a backup approach in cases where a multiple-group specification may be less convenient (for example, models with further covariates than just group dummies). In Section 5 we will discuss a similar choice between covariate specifications and multiple-group specifications for multigroup latent class models. There the relative preference turns out to be reversed, with the covariate specification being typically more convenient in practice.

Figure 11: Mplus input syntax for a multigroup latent trait model with 1 trait, in a covariate specification (Model N3C).

```
Title: LCAT_LT_N3CX
    LCAT: examples of multiple-group latent variable models
    Latant trait analysis, one trait, Model N3C
        Factor means depend on country, variances do not
        Non-equivalence of measurement in item "polinter", in both intercept and loading
        Covariate specification in Mplus
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-latent-traits
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind bel bul;
    Categorical = polmind polhard;
    Nominal = polinter;
Define:
    cut polhard(2);
    bel = (country==1);
    bul = (country==2);
Analysis:
    Estimator=ML;
    Type = Random; ! N3A2-N3C
!! This is not required but is usually advisable
    Starts = 20 10;
Model:
!! Basic measurement model and latent scale (all models)
    engage BY polhard* polmind;
    polinter ON engage;
    engage@1;
!!
! engage ON bel@0 bul@0; ! E0
engage ON bel bul; ! E1,E2,N2A2-N2C, N3A2-N3C
!!
!    polmind ON bel bul; ! N2A2
!    polhard ON bel bul; ! N2B
!    polinter ON bel bul; ! N2C
!!
    bel_en | bel XWITH engage; ! N3A2,N3B,N3C
    bul_en | bul XWITH engage; ! N3A,N3B,N3C
!    polmind ON bel bul bel_en bul_en; ! N3A2
!    polhard ON bel bul bel_en bul_en; ! N3B
    polinter ON bel bul bel_en bul_en; ! N3C
Savedata:
    File="tmp.dat";
```

Figure 12: Mplus input syntax for a multigroup latent trait model with 1 trait, in a multiple-group specification (Model N3C).

```
Title: LCAT_LT_N3CMG
    LCAT: examples of multiple-group latent variable models
    Latent trait analysis, one trait, Model N3C
        Factor means depend on country, variances do not
        Non-equivalence of measurement in item "polinter", in intercept and loading
        Multiple-group specification in Mplus
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-latent-traits
Data:
    File = ess4_3c.dat;
Variable:
    Names = idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind;
    Categorical = polmind polhard;
    Nominal = polinter;
    Classes = countryx(3);
    Knownclass= countryx (country=1 country=2 country=4);
Define: cut polhard(2);
Analysis:
    Type = Mixture;
    Estimator=ML;
    Algorithm = Integration;
!! This is not required but is usually advisable
     Starts = 20 10;
Model:
    %overall%
        engage BY polhard* polmind;
        polinter ON engage;
        engage@1;
!        [engage@0] (2); ! E0
!        [polinter#1](3); [polinter#2](4); [polinter#3](5); ! All but N2C,N3C,N6
!       [polmind$1](a1); [polmind$2](a2); [polmind$3](a3); [polmind$4](a4); ! N2A2,N3A2
    %countryx#1% ! E2,N2A-C,N3A-C
!        engage; ! E2
!        [polmind$1 polmind$2 polmind$3 polmind$4]; ! N2A,N3A
!       [polmind$1](b1); [polmind$2](b2); [polmind$3](b3); [polmind$4](b4); ! N2A2,N3A2
!        [polhard$1]; ! N2B,N3B
        [polinter#1 polinter#2 polinter#3]; ! N2C,N3C
!        engage BY polmind; ! N3A,N3A2
!        engage BY polhard; ! N3B
        polinter ON engage; ! N3C
    %countryx#2% ! E2,N2A-C,N3A-C
!        engage; ! E2
!        [polmind$1 polmind$2 polmind$3 polmind$4]; ! N2A,N3A
!       [polmind$1](c1); [polmind$2](c2); [polmind$3](c3); [polmind$4](c4); ! N2A2,N3A2
!        [polhard$1]; ! N2B,N3B
        [polinter#1 polinter#2 polinter#3]; ! N2C,N3C
!        engage BY polmind; ! N3A,N3A2
!        engage BY polhard; ! N3B
        polinter ON engage; ! N3C
!Model constraint: ! N2A2,N3A2
!    0=b1-a1+a2-b2; 0=b2-a2+a3-b3; 0=b3-a3+a4-b4; ! N2A2,N3A2
!    0=c1-a1+a2-c2; 0=c2-a2+a3-c3; 0=c3-a3+a4-c4; ! N2A2,N3A2
Savedata:
    File="tmp.dat"; Save=Cprobabilities; ! NOTE: need to include this, otherwise Mplus does not save
                         ! the known-class variable
```

4.2.2 Output

Figure 13 shows part of the Mplus output for one of the models above (N3C) fitted using the multiple-group specification (output for the corresponding covariate specification is not shown, because it would in practice be used less often; this output can be found on the LCAT website). The full output will again show estimates for every model parameter separately for each of the "latent classes", which are here the "known classes" defined by the groups, as stated in the input by the `Knownclass` option of the VARIABLE command, and numbered in the order listed there (here 1 for Belgium, 2 for Bulgaria and 3 for Cyprus). In Figure 13 all of the parameters are shown for the first group, and for the other two groups only those parameters which vary between groups are shown (here these are the mean of the latent trait, and the measurement parameters of the item *polinter*).

Different parameters are labelled as follows in the output. For the structural model we have

- `Means`: The means $\kappa^{(g)}$ of the latent trait. Here $\hat{\kappa}^{(1)} = -0.981$, $\hat{\kappa}^{(2)} = -0.408$, and $\hat{\kappa}^{(3)} = 0$ (constrained rather than estimated).
- `Variances`: Variances $\phi^{(g)}$ of the latent trait. Here this is is constrained to be $\phi^{(g)} = \phi = 1$ in every group.

and for the measurement model

- the slope coefficients (factor loadings):
  - `ENGAGE BY POLHARD` and `ENGAGE BY POLMIND` show the loadings for these observed items which are specified as categorical (i.e. ordinal), so these are the coefficients $\lambda_j^{(g)}$ in equation (36). Here $\hat{\lambda}_1^{(g)} = \hat{\lambda}_1 = 3.553$ for `polmind` (if we label it item $j = 1$) and $\hat{\lambda}_2^{(g)} = \hat{\lambda}_2 = -1.563$ for `polhard`.
  - `POLINTER#<level> ON ENGAGE` show the loadings for the nominal item *polinter*, i.e. the coefficients $\lambda_{jl}^{(g)}$ in equation (34), for `level`$= l = 1, \ldots, L-1 = 1, 2, 3$. In this model these vary by group, so that for example (labelling *polinter* as item $j = 3$) $\hat{\lambda}_{31}^{(1)} = 1.988$, $\hat{\lambda}_{31}^{(2)} = 2.460$, and $\hat{\lambda}_{31}^{(3)} = 2.855$.
    - Note that if we had specified the measurement model for *polinter* as `engage BY polinter#1 polinter#2 polinter#3`, these coefficients would appear in the output under `ENGAGE BY POLINTER#<level>`.
- and the intercept terms:
  - `Thresholds` are the intercepts $\tau_{jl}^{(g)}$ in the model (36) for the ordinal items. For example, `POLMIND$4` shows the estimate $\hat{\tau}_{14}^{(g)} = \hat{\tau}_{14} = 3.718$ for *polmind* (item $j = 1$).
  - `Intercepts` are the intercepts $\tau_{jl}^{(g)}$ in the model (34) for the nominal items. For example, `POLINTER#1` show the estimates of $\tau_{31}^{(g)}$ for *polinter* (item $j = 3$), e.g. $\hat{\tau}_{31}^{(2)} = -0.249$ for Bulgaria (country $g = 2$).

The output also has an entry for `Means of Categorical Latent Variables`. These simply specify the observed proportions of the groups (here countries), as parameters of a multinomial logistic model for the group. For example, here the sample proportion of observations from Belgium is $\exp(0.371)/[1 + \exp(0.371) + \exp(0.606)] = 0.338$.

Figure 14 shows the estimates for the same model, now as formatted by the `lcat` post-pocessing functions in R (see Section 7). In this display, the loadings for ordinal items (here *polhard* and *polmind*) are multiplied by -1 from the Mplus results, to align them better with the definition of the loadings for nominal items. Furthermore, estimates for the structural model are shown as differences from the estimates in the highest-numbered group, where the parameters are (in this model) fixed. Thus, for example, the '0' under the standard deviation of the latent trait (`ENGAGE`) for Belgium and Bulgaria indicate that the estimated (or in this case fixed) values of this parameter for these countries are equal to the value (1) for Cyprus.

Finally, let us check the interpretation of the fitted measurement model in the example of Tables 13 and 14:

- For the ordinal item `polmind`, the positive sign of the estimated coefficient 3.553 in the Mplus output implies that increasing the value of the latent trait η increases the probability of higher-numbered responses, which here indicate that a respondent finds it easy to make up his or her mind about politics. The same conclusion is (of course) reached from the `lcat` output where the coefficient is shown with the sign reversed as -3.553: this implies that increasing the trait decreases the probability of lower-numbered responses, i.e. the same interpretation.
- The binary item `polhard` is here specified in Mplus as ordinal, so the interpretation of the sign of its loading is similar to that of `polmind`. Here the coefficient In Mplus output is -1.563. This implies that increasing the latent trait decreases the probability of the higher-numbered response to `polhard`, which corresponds to a respondent who finds politics complicated to understand.
- For the nominal item `polinter`, the coefficients are log odds ratios for its levels 1–3 relative to the highest level 4 which corresponds to the lowest level of interest in politics. Here the coefficients for Belgium (country 1) are 1.988, 1.277 and 0.587 for levels 1, 2, and 3 respectively. In the other countries these estimates have different values because this item is specified as non-equivalent across the countries, but all of them are positive and in the same order of size where the coefficient for level 1 is largest that of level 3 smallest. These values imply that increasing the value of the latent trait increases the probability that a respondent chooses a lower-numbered response to the item, i.e. indicates a high level of interest in politics.

In short, all the measurement models agree about the qualitative interpretation of the latent trait. This is that high values of the trait correspond to an individual who is interested in politics, finds politics easy to understand, and finds it easy to make up their mind about political issues. We might label this as "political interest" for short. The estimated means ($\hat{\kappa}^{(g)}$) of this trait are -0.981, -0.408 and (fixed) 0 in Belgium, Bulgaria and Cyprus respectively, on a scale where the individual-level standard deviation is 1. These estimates thus indicate that the average level of interest in politics is highest in Cyprus and lowest in Belgium.

Figure 13: Part of Mplus output for a multigroup latent trait model with 1 trait, in a multiple-group specification (Model N3C).

```
                                                          Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value
Latent Class 1
 ENGAGE   BY
    POLHARD           -1.563      0.071    -22.065      0.000
    POLMIND            3.553      0.325     10.930      0.000
 POLINTER#1 ON
    ENGAGE             1.988      0.189     10.500      0.000
 POLINTER#2 ON
    ENGAGE             1.277      0.123     10.426      0.000
 POLINTER#3 ON
    ENGAGE             0.587      0.107      5.498      0.000
 Means
    ENGAGE            -0.981      0.047    -20.994      0.000
 Intercepts
    POLINTER#1         0.930      0.211      4.411      0.000
    POLINTER#2         2.198      0.183     12.008      0.000
    POLINTER#3         1.339      0.173      7.750      0.000
 Thresholds
    POLHARD$1         -0.206      0.064     -3.194      0.001
    POLMIND$1         -7.726      0.594    -13.011      0.000
    POLMIND$2         -3.869      0.324    -11.943      0.000
    POLMIND$3         -0.395      0.134     -2.946      0.003
    POLMIND$4          3.718      0.291     12.795      0.000
 Variances
    ENGAGE             1.000      0.000    999.000    999.000
Latent Class 2
 POLINTER#1 ON
    ENGAGE             2.460      0.180     13.642      0.000
 POLINTER#2 ON
    ENGAGE             1.473      0.114     12.940      0.000
 POLINTER#3 ON
    ENGAGE             0.736      0.095      7.740      0.000
 Means
    ENGAGE            -0.408      0.042     -9.618      0.000
 Intercepts
    POLINTER#1        -0.249      0.152     -1.634      0.102
    POLINTER#2         1.243      0.115     10.831      0.000
    POLINTER#3         0.750      0.107      6.977      0.000
Latent Class 3
 POLINTER#1 ON
    ENGAGE             2.855      0.238     11.981      0.000
 POLINTER#2 ON
    ENGAGE             1.730      0.167     10.340      0.000
 POLINTER#3 ON
    ENGAGE             0.740      0.121      6.095      0.000
 Means
    ENGAGE             0.000      0.000    999.000    999.000
 Intercepts
    POLINTER#1        -0.651      0.195     -3.334      0.001
    POLINTER#2         0.560      0.126      4.426      0.000
    POLINTER#3         0.789      0.117      6.759      0.000
Categorical Latent Variables
 Means
    COUNTRYX#1         0.371      0.037      9.956      0.000
    COUNTRYX#2         0.606      0.036     16.993      0.000
```

Figure 14: Output from the `lcat` post-processing functions in R for a multigroup latent trait model with 1 trait fitted in Mplus (Model N3C). Mplus output for the same model is shown in Figure 13.

```
LCAT output
Mplus file:  lcat_lt_n3cmg
Latent trait model, with 1 latent trait: ENGAGE

3 categorical items:
 Name     Level   Categories Invariant
 POLHARD  Ordinal 2          Yes
 POLMIND  Ordinal 5          Yes
 POLINTER Nominal 4          No
Multiple group model, with 3 groups:
BEL  BUL  CYP

Model estimates:
N = 5200        parameters = 27                log-likelihood = -15895.69
AIC = 31845.38  BIC = 32022.4   Delta =  0.133 ( 0.11 - 0.153 across groups )
% of 2-way marginal residuals > 4: 55 ( 39 - 50 across groups )

Models for the the latent traits:
  - difference to the reference group ( CYP )

Trait  ENGAGE :
      Mean sd
BEL -0.981  0
BUL -0.408  0
CYP  0.000  1

Parameters of the measurement model:
'$' indicates intercept of an ordinal logistic model,
and '#' of a multinomial logistic model.

Positive loading of a trait indicates that higher values of the trait
correspond to higher probabilities lower-numbered categories in ordinal model
and higher probability of a category relative to the highest-numbered category
in multinomial model.

For items that are invariant across groups:
          Constant ENGAGE
POLHARD$1   -0.206  1.563

          Constant ENGAGE
POLMIND$1   -7.726 -3.553
POLMIND$2   -3.869 -3.553
POLMIND$3   -0.395 -3.553
POLMIND$4    3.718 -3.553

For items that are not invariant across groups:
               Constant ENGAGE
POLINTER#1.BEL    0.930  1.988
POLINTER#2.BEL    2.198  1.277
POLINTER#3.BEL    1.339  0.587
POLINTER#1.BUL   -0.249  2.460
POLINTER#2.BUL    1.243  1.473
POLINTER#3.BUL    0.750  0.736
POLINTER#1.CYP   -0.651  2.855
POLINTER#2.CYP    0.560  1.730
POLINTER#3.CYP    0.789  0.740
```

4.3 2-trait multigroup models in Mplus

4.3.1 Input

For an example of latent trait models with two traits, we consider again data from the ESS example summarised in Appendix A. The first of the traits is *political interest*, measured by the three indicators `polinter`, `polhard` and `polmind` in the same way as in Section 4.2 above. The second latent trait is one we label *trust in political institutions*. It is measured by the three indicators `tparl`, `tpolitic` and `tparties`. All of them are here dichotomised by combining their original levels 0–4 (new level 1, corresponding to low level of trust) and 5–10 (2, high level of trust).

The structural models for the two traits are as specified in Section 4.1. As noted there, the association between the traits may be specified either symmetrically, in what we call the covariance specification, or with a regression specification where one trait is treated as an explanatory variable for the other. Here we consider both possibilities. When a regression specification is used, political interest is specified as explanatory for trust.

We consider here only cases where each item y_j measures only one of the latent traits. The measurement models are then of the forms stated in Section 4.1, except that we replace $\eta^{(g)}$ in the formulas with $\eta_1^{(g)}$ or $\eta_2^{(g)}$. The more general case where some items measure both traits (i.e. have non-zero loading parameters for both of them) is obtained with fairly obvious modifications of the Mplus syntax which are not discussed here.

The models can again be specified in Mplus using either a covariate specification or a multiple-group specification, as discussed in Section 4.2.1. As the multiple-group specification is more flexible and generally faster, we consider only it here. Examples of the covariate specification are given at the LCAT website (`http://stats.lse.ac.uk/lcat/`), as are full syntax files for the multiple-group specification for all the cases discussed below.

We consider a range of different model specifications, in parallel for the covariance and regression specifications. These are summarised in Table 4. Mplus input for one model (E3) is shown, in Figure 15 for the covariance specification and in Figure 16 for the regression specification. Syntax for the other cases is again obtained by commenting and uncommenting lines from these, as indicated by comments in the syntax. The cases differ mainly in what constraints are imposed on the structural model, i.e. which parameters of this model do and do not vary across groups. The specification of the measurement model is essentially similar to the one-trait case. All but one of the examples (N4) have complete measurement equivalence. The measurement model is then specified by the lines

```
Model:
    %overall%
        engage BY polhard* polmind;
        polinter ON engage;
        poltrust BY tparl* tpolitic tparties;
        [polinter#1](2); [polinter#2](3); [polinter#3](4);
```

Non-equivalence models are obtained by extending these in the same ways as for 1-trait models in Section 4.2.1.

Table 4: Summary of the 2-trait models considered in Section 4.3. See equations (25)–(28) for the notation, and Figures 15 and 16 for full input syntax for Model E3 with the covariance and regression specifications respectively.

		Parameters that are constant across groups, in two different specifications:	
	Summary of structural model:	Covariance	Regression
E1	All parameters vary across groups	None	None
E2	Trait variances do not vary	(ϕ_{11}, ϕ_{22})	(ϕ_{11}, ψ)
E3	(Conditional) variance of η_2 and trait association do not vary	(ϕ_{12}, ϕ_{22})	(γ_1, ψ)
E4	η_2 is marginally or conditionally independent of group, and variance of η_1 does not depend on group	$(\phi_{11}, \phi_{12}, \phi_{22}, \kappa_2)$	$(\phi_{11}, \gamma_1, \psi, \gamma_0)$
E5	η_2 and η_1 are independent given group	$\phi_{12} = 0$	$\gamma_1 = 0$
N4	Like E4, but also non-equivalence of measurement in intercept of one item		

The models listed in Table 4 are specified as follows:

- **E1**: Unconstrained structural model, where all of its parameters vary across groups.
  - Covariance specification (E1C): The relevant lines of syntax are

    ```
    Model:
        %overall%
            engage@1; poltrust@1; [engage@0]; [poltrust@0];
            poltrust WITH engage;
        %countryx#1% !
            engage; poltrust; [engage]; [poltrust];
            poltrust WITH engage;
    ```

    plus similar lines for `countryx#2` as for `countryx#1` (this is the case in all of the examples below). Here the `%overall%` part imposes the necessary identifiability conditions on the means and variances of the latent traits, and the group-specific specifications `%countryx#1%` and `%countryx#2%` then free these parameters to be estimated in all but one group (here group 3). The `poltrust WITH engage;` line requests the covariance between the traits to be estimated, and repeating this line under the group-specific commands means that the covariance is estimated separately in each group.
  - Regression specification (E1R): This is similar to E1C, except that the line `poltrust WITH engage;` is replaced with `poltrust ON engage;` wherever it appears, to parametrise the association between the traits through a regression model rather than a covariance. Note that `[poltrust];` and `poltrust;` will then refer to the intercept and residual variance in the regression for `poltrust` given `engage` rather than marginal mean and variance of `poltrust`.

- **E2**: Structural variance parameters (marginal or residual variances) are constant across groups, and fixed at 1. In both specifications, this is obtained by deleting in the syntax for model E1 the variance specifications `engage;` and `poltrust;` from the group-specific models `%countryx#1%` and `%countryx#2%`.
- **E3**: The association between the traits and the marginal (in E3C) or conditional (in E3R) variance of `poltrust` are constant across groups.
  - Covariate specification (E3C): This is obtained by deleting in the syntax for E1C `poltrust WITH engage;` and `poltrust;` from the group-specific models `%countryx#1%` and `%countryx#2%`.
  - Covariate specification (E3R): This is obtained by deleting in the syntax for E1R `poltrust ON engage;` and `poltrust;` from the group-specific models `%countryx#1%` and `%countryx#2%`.
- **E4**: Marginal variance of `engage` and the marginal (in E4C) or conditional (in E4R) distribution of `poltrust` are constant across groups. In both specifications, this is obtained by leaving only `[engage];`, i.e. the marginal mean of `engage`, specified under the group-specific models `%countryx#1%` and `%countryx#2%`.
- **E5**: Like E1, except that the association parameter between the factors is 0 in all groups. This means that `engage` and `poltrust` are conditionally independent given group.
  - Covariance specification (E5C): Like E1C except that the covariance between the traits is listed only under `%overall%` specification, and set to 0 with `poltrust WITH engage@0;`.
  - Regression specification (E5C): Like E1R except that the regression coefficient between the traits is listed only under `%overall%` specification, and set to 0 with `poltrust ON engage@0;`.
- **N4**: Like E4, but in addition the measurement intercept of the binary item `polhard` varies across groups. This is done in the same way for both specifications, and in the same way as for one-trait models, by adding the line `[polhard$1];` under the the group-specific models `%countryx#1%` and `%countryx#2%`. Other instances of non-equivalence of measurement are also specified in the same way as for one-trait models, for which examples are given in Section 4.2.1.

It should be noted that for models E1 and E5 the two specifications are equivalent, i.e. specify the same model. In the other cases the covariance and regression specifications imply somewhat analogous but not identical models because the cross-group constraints are not applied to exactly equivalent parameters.

Figure 15: Mplus input syntax for a multigroup latent trait model with 2 traits (Model E3). The model is parametrised using the covariate specification, and fitted in Mplus using its multiple-group specification.

```
Title: LCAT_LT2_E3CMG
    LCAT: examples of multiple-group latent variable models
    Latent trait analysis, two traits, Model E3CMG
        Covariance specification of the model
        Covariance between traits and one (residual) trait variance
            do not vary between countries.
        Measurement equivalence in all items
        Multiple-group specification in Mplus
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-latent-traits
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind tparl tpolitic tparties;
    Categorical = polmind polhard tparl tpolitic tparties;
    Nominal = polinter;
    Classes = countryx(3);
    Knownclass= countryx (country=1 country=2 country=4);
Define:
    cut polhard(2); cut tparl-tparties(4);
Analysis:
    Type=Mixture;
    Estimator=ML;
    Algorithm = Integration;
!! These are not required but may often be necessary, especially Starts
     Starts = 20 10;
!    Stiterations = 20; !    Integration = 20;
Model:
    %overall%
        engage BY polhard* polmind;
        polinter ON engage;
        poltrust BY tparl* tpolitic tparties;
        engage@1; poltrust@1;
        [engage@0]; [poltrust@0];
        [polinter#1](2); [polinter#2](3); [polinter#3](4);
!        [poltrust@0] (1); ! N4CMG
      poltrust WITH engage; ! N4CMG,E4CMG,E3CMG,E2CMG,E1CMG
!      poltrust WITH engage@0; ! E5CMG
    %countryx#1% !
        engage; ! E5CMG,E3CMG,E1CMG
!        poltrust; ! E5CMG,E1CMG
        [engage];
        [poltrust]; ! E5CMG,E3CMG,E2CMG,E1CMG
!        poltrust WITH engage; ! E2CMG,E1CMG
!        [polhard$1]; ! N4CMG
   %countryx#2% !
        engage; ! E5CMG,E3CMG,E1CMG
!        poltrust; ! E5CMG,E1CMG
        [engage];
        [poltrust]; ! E5CMG,E3CMG,E2CMG,E1CMG
!        poltrust WITH engage; ! E2CMG,E1CMG
!        [polhard$1]; ! N4CMG
```

Figure 16: Mplus input syntax for a multigroup latent trait model with 2 traits (Model E3). The model is parametrised using the regression specification, and fitted in Mplus using its multiple-group specification.

```
Title: LCAT_LT2_E3RMG
    LCAT: examples of multiple-group latent variable models
    Latent trait analysis, two traits, Model E3RMG
        Regression specification of the model
        Regression coefficient between traits and one (residual) trait variance
            do not vary between countries.
        Measurement equivalence in all items
        Multiple-group specification in Mplus
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/?resources=computing-latent-traits
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind tparl tpolitic tparties;
    Categorical = polmind polhard tparl tpolitic tparties;
    Nominal = polinter;
    Classes = countryx(3);
    Knownclass= countryx (country=1 country=2 country=4);
Define:
    cut polhard(2); cut tparl-tparties(4);
Analysis:
    Type=Mixture;
    Estimator=ML;
    Algorithm = Integration;
!! These are not required but may often be necessary, especially Starts
     Starts = 20 10;
!    Stiterations = 20; !    Integration = 20;
Model:
    %overall%
        engage BY polhard* polmind;
        polinter ON engage;
        poltrust BY tparl* tpolitic tparties;
        engage@1; poltrust@1;
        [engage@0]; [poltrust@0];
        [polinter#1](2); [polinter#2](3); [polinter#3](4);
!        [poltrust@0] (1); ! N4RMG
      poltrust ON engage; ! N4RMG,E4RMG,E3RMG,E2RMG,E1RMG
!      poltrust ON engage@0; ! E5RMG
    %countryx#1% !
        engage; ! E5RMG,E3RMG,E1RMG
!        poltrust; ! E5RMG,E1RMG
        [engage];
        [poltrust]; ! E5RMG,E3RMG,E2RMG,E1RMG
!        poltrust ON engage; ! E2RMG,E1RMG
!        [polhard$1]; ! N4RMG
   %countryx#2% !
        engage; ! E5RMG,E3RMG,E1RMG
!        poltrust; ! E5RMG,E1RMG
        [engage];
        [poltrust]; ! E5RMG,E3RMG,E2RMG,E1RMG
!        poltrust ON engage; ! E2RMG,E1RMG
!        [polhard$1]; ! N4RMG
```

4.3.2 Output

Structure of the Mplus output for 2-trait multigroup models is essentially the same as for 1-trait models. When the model is fitted using the multiple-group specification, all parameter estimates are again listed for every group (i.e. every "Known class" in Mplus).

Figure 17 shows part of the Mplus output for Model E3, from both the covariance (E3C) and regression (E3R) specifications. Only estimates of the parameters of the structural model are shown, and only for two groups (Bulgaria and the reference group Cyprus). Parameters of measurement models are displayed exactly as for 1-trait models, so they are not shown here. Fuller output for these models, from the lcat post-processing functions in R (see Section 7) are shown in Figures 18 (for E3C) and 19 (for E3R).

Below we number the factors engage as η_1 and poltrust as η_2, and the two groups shown in the output as Bulgaria 2 and Cyprus 3. The labelling of parameters in the Mplus output is the same as for 1-trait models. For the covariance specification, the new element in the output here is the covariance between the traits, which is labelled as <trait> WITH <trait>, so here poltrust WITH engage. Its estimate is here $\hat{\phi}_{12}^{(g)} = \phi_{12} = 0.081$, which is in this model constrained to be equal across the groups.

The same labelling of the factor means and variances is used in the regression formulation for any trait which is a not response variable to another trait, such as engage (η_1) here. For a trait which is a response variable, the following labelling is used:

- Intercepts: Regression intercepts $\gamma_0^{(g)}$, e.g. $\hat{\gamma}_0^{(2)} = -1.566$.
- <response_factor> ON <explanatory_factor>: Regression coefficients $\gamma_1^{(g)}$, here $\hat{\gamma}_1^{(g)} = \hat{\gamma}_1 = 0.138$.
- Residual Variances: Residual variances $\psi^{(g)}$ in the structural regression model, here $\psi^{(g)} = \psi = 1$.

Finally, we note two features of the lcat function output, examples of which are shown in Figures 18 and 19. First, to display the structural model the traits are ordered in such a way that the first to be shown is not a response to the other trait, and means and standard deviations of this trait across the groups are shown. For the second trait, the output shows either its marginal mean and standard deviation and covariance with the first trait (for the covariance specification) or intercept, regression coefficient and residual standard deviation of the model for it given the first trait (for the regression specification). The different cases are identified by labels of the parameters in the table. Second, the measurement model is shown in the form of fitted probabilities of responses to the items at 5 values of the latent trait, at its mean (in one group) and ± 2 standard deviations from the mean (although the estimated intercept and loading parameters of the measurement model can also be requested instead). Here these probabilities indicate that the engage trait is defined in such a way that high levels of it indicate high levels of interest in politics, and poltrust in such a way that high levels correspond to high levels of trust in political intitutions. The estimated association between these traits is positive in these countries.

Figure 17: Part of Mplus output for a multigroup latent trait model with 2 traits (Model E3). Output for both the covariance specification (E3C) and regression specification (E3R) are shown. Only estimates of the parameters of the structural model are shown, for two groups (Bulgaria ["Latent Class 2"] and the reference group Cyprus [3]) only.

```
-------------------------------------------------------------
Covariance specification (Model E3C):
-------------------------------------------------------------
Latent Class 2
 POLTRUST WITH
    ENGAGE             0.081      0.016      4.978      0.000
 Means
    ENGAGE            -0.315      0.038     -8.382      0.000
    POLTRUST          -1.601      0.060    -26.635      0.000
 Variances
    ENGAGE             0.592      0.042     14.057      0.000
    POLTRUST           1.000      0.000    999.000    999.000

Latent Class 3
 POLTRUST WITH
    ENGAGE             0.081      0.016      4.978      0.000
 Means
    ENGAGE             0.000      0.000    999.000    999.000
    POLTRUST           0.000      0.000    999.000    999.000
 Variances
    ENGAGE             1.000      0.000    999.000    999.000
    POLTRUST           1.000      0.000    999.000    999.000
-------------------------------------------------------------
Regression specification (Model E3R):
-------------------------------------------------------------
Latent Class 2
 POLTRUST   ON
    ENGAGE             0.138      0.026      5.228      0.000
 Means
    ENGAGE            -0.319      0.037     -8.536      0.000
 Intercepts
    POLTRUST          -1.566      0.065    -24.061      0.000
 Variances
    ENGAGE             0.585      0.042     13.965      0.000
 Residual Variances
    POLTRUST           1.000      0.000    999.000    999.000

Latent Class 3
 POLTRUST   ON
    ENGAGE             0.138      0.026      5.228      0.000
 Means
    ENGAGE             0.000      0.000    999.000    999.000
 Intercepts
    POLTRUST           0.000      0.000    999.000    999.000
 Variances
    ENGAGE             1.000      0.000    999.000    999.000
 Residual Variances
    POLTRUST           1.000      0.000    999.000    999.000
-------------------------------------------------------------
```

Figure 18: Output from the `lcat` post-processing functions in R for a multigroup latent trait model with 2 traits (with a covariance specification, Model E3C) fitted in Mplus.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_lt2_e3cmg
Latent trait model, with 2 latent traits: ENGAGE POLTRUST

6 categorical items:
 Name     Level   Categories Invariant
 POLHARD  Ordinal 2          Yes
 POLMIND  Ordinal 5          Yes
 TPARL    Ordinal 2          Yes
 TPOLITIC Ordinal 2          Yes
 TPARTIES Ordinal 2          Yes
 POLINTER Nominal 4          Yes

Multiple group model, with 3 groups:
CYP  BEL  BUL
Model estimates:
N = 5205        parameters = 26                 log-likelihood = -22275.3
AIC = 44602.6   BIC = 44773.1   Delta =  0.21 ( 0.186 - 0.228 across groups )
% of 2-way marginal residuals > 4: 38 ( 42 - 49 across groups )
Sum of 2-way marginal residuals large, out of 15 pairs: 12 ( 13 - 15 across groups )

Models for the the latent traits:
Trait  ENGAGE :
      Mean    sd
CYP  0.000 1.000
BEL -0.719 0.699
BUL -0.315 0.769

Trait  POLTRUST :
      Mean cov.ENGAGE sd
CYP  0.000      0.081  1
BEL -0.272      0.081  1
BUL -1.601      0.081  1

Measurement probabilities
conditional on each latent trait at m+(-2,-1,0,1,2)*sd
where m and sd are the mean and standard deviation of the latent trait in group CYP :

For items that are invariant across groups:
Given trait ENGAGE :
                        m-2sd   m-1sd   mean    m+1sd   m+2sd
        POLHARD#1       0.018   0.108   0.440   0.836   0.971
        POLHARD#2       0.982   0.892   0.560   0.164   0.029

        TPARL#1         0.210   0.210   0.210   0.210   0.210
        TPARL#2         0.790   0.790   0.790   0.790   0.790
[... some omitted ...]

Given trait POLTRUST :
                        m-2sd   m-1sd   mean    m+1sd   m+2sd
        POLHARD#1       0.440   0.440   0.440   0.440   0.440
        POLHARD#2       0.560   0.560   0.560   0.560   0.560

        TPARL#1         0.977   0.769   0.210   0.021   0.002
        TPARL#2         0.023   0.231   0.790   0.979   0.998
[... some omitted ...]
--------------------------------------------------------------------------------
```

Figure 19: Output from the `lcat` post-processing functions in R for a multigroup latent trait model with 2 traits (with a regression specification, Model E3C) fitted in Mplus.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_lt2_e3rmg
Latent trait model, with 2 latent traits: ENGAGE POLTRUST

6 categorical items:
 Name     Level   Categories Invariant
 POLHARD  Ordinal 2          Yes
 POLMIND  Ordinal 5          Yes
 TPARL    Ordinal 2          Yes
 TPOLITIC Ordinal 2          Yes
 TPARTIES Ordinal 2          Yes
 POLINTER Nominal 4          Yes

Multiple group model, with 3 groups:
CYP  BEL  BUL

Model estimates:
N = 5205        parameters = 26                 log-likelihood = -22273.64
AIC = 44599.29  BIC = 44769.78  Delta =  0.21 ( 0.186 - 0.226 across groups )
% of 2-way marginal residuals > 4: 38 ( 44 - 49 across groups )
Sum of 2-way marginal residuals large, out of 15 pairs: 12 ( 13 - 15 across groups )

Models for the the latent traits:
Trait  ENGAGE :
      Mean    sd
CYP  0.000 1.000
BEL -0.721 0.693
BUL -0.319 0.765

Trait  POLTRUST :
    Intercept b.ENGAGE resid.sd
CYP     0.000    0.138        1
BEL    -0.174    0.138        1
BUL    -1.566    0.138        1

Measurement probabilities conditional on each latent trait at m+(-2,-1,0,1,2)*sd
where m and sd are the mean and standard deviation of the latent trait in group CYP :

For items that are invariant across groups:
Given trait ENGAGE :
                        m-2sd   m-1sd   mean    m+1sd   m+2sd
        POLHARD#1       0.018   0.108   0.443   0.839   0.972
        POLHARD#2       0.982   0.892   0.557   0.161   0.028

        TPARL#1         0.209   0.209   0.209   0.209   0.209
        TPARL#2         0.791   0.791   0.791   0.791   0.791
[... some omitted ...]

Given trait POLTRUST :
                        m-2sd   m-1sd   mean    m+1sd   m+2sd
        POLHARD#1       0.443   0.443   0.443   0.443   0.443
        POLHARD#2       0.557   0.557   0.557   0.557   0.557

        TPARL#1         0.977   0.770   0.209   0.020   0.002
        TPARL#2         0.023   0.230   0.791   0.980   0.998
[... some omitted ...]
--------------------------------------------------------------------------------
```

5 Latent class models for multiple groups

5.1 General specification of the models

By a *multigroup latent class model*, we mean a model of the general form (4) where the latent variables $\boldsymbol{\eta}$ as well as the observed items $\mathbf{y}$ are categorical variables.

Turning first to the structural model of a latent class model, we will consider only situations with a single latent variable. Models with several categorical latent variables are possible but are not discussed here. Let the latent variable in group $g = 1, \ldots, G$ be denoted $\eta^{(g)}$ as before. Unlike in factor analysis and latent trait models, however, now $\eta^{(g)}$ is a categorical variable, with C possible values (*latent classes*) $c = 1, \ldots, C$. The number of classes C is treated as fixed when estimating the model, and the value of C for the model to be used for interpretation is determined by comparing the goodness of fit and interpretability of fitted models with different values of C.

The structural model for $\eta^{(g)}$ can be formulated as a multinomial logistic model given the group g, as

$$P(\eta^{(g)} = c) \equiv \alpha_c^{(g)} = \frac{\exp(\kappa_{0c} + \kappa_c^{(g)})}{\sum_{c'=1}^{C} \exp(\kappa_{0c'} + \kappa_{c'}^{(g)})} \tag{41}$$

for latent classes $c = 1, \ldots, C$ and groups $g = 1, \ldots, G$, where the κ-quantities are model parameters. In Mplus, $\kappa_{0C} = 0$ and $\kappa_C^{(1)} = \cdots = \kappa_C^{(G)} = 0$, i.e. the reference class for η is the largest-numbered latent class. In the Mplus multiple-group specification (discussed below), the highest-numbered group is also the reference level for the group variable, i.e. $\kappa_c^{(G)} = 0$ for all c. In the covariate specification (also discussed below) we can choose the reference group by omitting the corresponding group dummy variable from the model.

Turning next to the measurement models of a latent class model, these are of the same form as the measurement models of a latent trait model, but with a categorical rather than continuous explanatory variable $\eta^{(g)}$. We repeat them here briefly in a notation tailored to the latent class case, and refer the reader to Section 4.1 for a more detailed discussion of the measurement models.

As in Section 4.1, $y_j^{(g)}$ denote the observed item $j = 1, \ldots, p$ in group $g = 1, \ldots, G$. Different items are again always assumed to be conditionally independent of each other given the latent class variable $\eta^{(g)}$, so we can without loss of generality describe the measurement models for one item at a time. Each item $y_j^{(g)}$ is again categorical, with L possible levels (*categories*) $l = 1, \ldots, L$ (different items can have different numbers of levels, so we should write L_j; we omit the subscript to simplify the notation). The item may be either *ordinal*, when the levels are taken to be ordered from 1 to L, or *nominal* when they are not ordered. When $L = 2$, an item is *binary* (dichotomous), which can be equivalently treated as either ordinal or nominal. In both ordinal and nominal cases we refer to levels 1 and L as the "lowest" and "highest" levels of an item respectively.

The most common choice is to assume that the items are nominal, in which case the measurement model for item $j = 1, \ldots, p$ is the multinomial logistic model

$$\pi_{jl}^{(g)}(c) \equiv P(y_j^{(g)} = l | \eta^{(g)} = c) = \frac{\exp(\tau_{jlc} + \lambda_{jlc}^{(g)})}{\sum_{c'=1}^{C} \exp(\tau_{jlc'} + \lambda_{jlc'}^{(g)})} \tag{42}$$

for $l = 1, \ldots, L$ and $c = 1, \ldots, C$, where for identifiability $\tau_{jLc} = \lambda_{jLc}^{(g)} = 0$ for all c, g and $\lambda_{jlc}^{(g)} = 0$ for one selected reference group g for all l, c.

An item $y_j^{(g)}$ can also be treated as ordinal. In this case, the measurement model is the ordinal logistic model

$$\nu_{jl}^{(g)}(c) \equiv P(y_j^{(g)} \leq l | \eta^{(g)} = c) = \frac{\exp(\tau_{jlc} - \lambda_{jc}^{(g)})}{1 + \exp(\tau_{jlc} - \lambda_{jc}^{(g)})} \tag{43}$$

for $l = 1, \ldots, L-1$, where for one group g we have $\lambda_{jc}^{(g)} = 0$ for all c. From this we can also get the probabilities of the individual response categories as $\pi_{jl}^{(g)}(c) = \nu_{jl}^{(g)}(c) - \nu_{j,l-1}^{(g)}(c)$ for $j = 1, \ldots, L$, where $\nu_{j0}(c) = 0$ and $\nu_{jL}(c) = 1$.

The τ and λ-parameters in (42) and (43) are not the same parameters, but we use the same notation to highlight their similar roles. In both cases measurement equivalence or non-equivalence is determined by the λ-parameters, so that item j is equivalent across the groups if $\lambda_{jlc}^{(g)} = 0$ for all l, c, g in (42) and if $\lambda_{jc}^{(g)} = 0$ for all c, g in (43).

Model (43) is used for ordinal items in Mplus. We note that this is actually less parsimoniously specified than a conventional ordinal logistic model, because it specifies the association of latent class and outcome through the $C \times (L-1)$ parameters τ_{jlc} and not through $(L-1)+(C-1)$ parameters of the form $\tau_{jl0} - \tau_{jc} \equiv \tau_{jlc}$ as would be more conventional. This then implies that when $\lambda_{jc}^{(g)} = 0$, (43) is in fact equivalent to (42) with $\lambda_{jlc}^{(g)} = 0$. In other words, when the measurement model of an item in a latent class model is equivalent across groups, specifying the item as nominal or ordinal in Mplus both actually result in the same (nominal) measurement model.

Finally, we note that for a multigroup latent class model, the probability distribution (1) of the observed items given group for a single subject is given by

$$P(\mathbf{y}^{(g)} = \mathbf{l}) = \sum_{c=1}^{C} \left[\prod_{j=1}^{p} P(y_j^{(g)} = l_j | \eta^{(g)} = c) \right] P(\eta^{(g)} = c) \tag{44}$$

where $\mathbf{l} = (l_1, \ldots, l_p)$ is the observed set of values for the items $\mathbf{y}$. The fact that (44) is a sum, rather than an intractable integral as for latent trait models, somewhat simplifies estimation for latent class models.

5.2 Mplus input

We use again the subset of data from the European Social Survey that is described in Appendix A. It involves respondents from three groups (countries), Belgium, Bulgaria and Cyprus.

Example syntax for one latent class model with $C = 3$ latent classes is given in Figures 20 and 21, separately for a covariate specification and a multiple-group specification. Commands for the other examples are obtained by commenting in and out lines from this syntax, as discussed below. The syntax for all the examples is also available at the LCAT website (`http://stats.lse.ac.uk/lcat/`).

We discuss one-trait models for 3 of the observed items, *polinter* (4 categories), *polhard* (2 categories) and *polmind* (5 categories). These are the same variables that were used for examples of 1-trait latent trait models in Section 4.2, and we refer the reader to page 39 for an explanation of them. In the examples below, we consider some latent class models with equivalence and some with partial or full non-equivalence. The items *polmind* and *polhard* are (except in Model N5) specified to have equivalent measurement models and are always treated as nominal. For the variable *polinter* we show examples of both equivalent and non-equivalent models, and of ordinal and nominal specifications.

A *single-group* latent class model is obtained in Mplus by specifying

- the observed items as categorical under `Variables:` in the same way as for latent trait models (see examples in Figures 20 and 21);
- the latent class variable and the number of classes with the `Variable: Classes` command, e.g. in the examples below
  `Variable:  Classes=class(3);`
  This states that the latent class variable will be labelled "class", and will have 3 latent classes.
- `Analysis:  Type=Mixture; Estimator=ML;`

For a *multigroup* model, this is then further modified. In Mplus this can be done in two quite different ways, which we refer to as the "multiple-group specification" and the "covariate specification". This parallels the similar situation for latent trait models, which was discussed in Section 4.2.1. Here, however, the preference ordering of the two approaches is reversed: for latent class models, the covariate specification is faster and more flexible and thus usually preferred. We describe it first below, before explaining the multiple-group specification more briefly.

We consider seven models to illustrate various types of parameter constraints. These cases are summarised in Table 5. Some of the cases differ from each other only in whether the item *polinter* is treated as a nominal or ordinal variable.

Consider first these models in the covariate specification. This requires that dummy variables for all but one of the groups are available. Here we need dummy variables for 2 of the 3 countries, which we take to be Bulgaria and Cyprus, leaving Belgium (which we label as group $g = 1$) as the reference country. The dummy variables may be included in the input data set, or created (here from the variable `country` in the data) within Mplus, here with

```
Define:
    bul = (country==2);
    cyp = (country==4);
```

Table 5: Summary of the latent class models considered in Sections 5.2 and 5.3. See equations (41)–(43) for the notation, and Figures 20 and 21 for full input syntax for Model N4.

	Key features of the model:
Models with measurement equivalence across groups in all items	
E0	Latent class probabilities equal across groups ($\kappa_c^{(g)} = 0$ for all groups g)
E1	Latent class probabilities vary across groups ($\kappa_c^{(g)}$ varies across groups g)
Models with non-equivalence of measurement in one item j, here always polinter *In all of these, $\kappa_c^{(g)}$ vary across groups.*	
N1	*polinter* is ordinal, $\lambda_{jc}^{(g)} = \lambda_j^{(g)}$ for all g (direct effect of group on item)
N2	*polinter* is nominal, $\lambda_{jlc}^{(g)} = \lambda_{jl}^{(g)}$ for all l, g (direct effect of group on item)
N3	*polinter* is ordinal, $\lambda_{jc}^{(g)}$ unconstrained (group-class interaction on item)
N4	*polinter* is nominal, $\lambda_{jlc}^{(g)}$ unconstrained (group-class interaction on item)
N5	Non-equivalence of measurement for all items, i.e. a 3-class model fitted separately for each group

The models in Table 5 which have full measurement equivalence are then obtained with the following specifications under the `Model:` command:

- **E0**:

```
%overall%
     class ON bul@0 cyp@0;
```

  This sets $\kappa_c^{(2)} = \kappa_c^{(3)} = 0$ in (41).

- **E1**:

```
%overall%
     class ON bul cyp;
```

  This lets $\kappa_c^{(2)}$ and $\kappa_c^{(3)}$ in (41) to be estimated freely.

As noted in Section 5.1 above, for both of these models Mplus will use a nominal measurement model for all items, irrespective of whether they are specified as nominal or ordinal under the `Variable:` command.

Next, for the models where the class probabilities vary freely and there is non-equivalence of measurement in the item *polinter*, the structural model is specified as for E1 above, and the measurement model is specified as follows:

- **N1**: `Variable: Categorical=polinter;` to specify *polinter* as ordinal, and then under `Model:`

```
%overall%
     polinter ON bul cyp;
```

This lets $\lambda_{jc}^{(g)} = \lambda_j^{(g)}$ in (43) vary between the groups g, but in the same way for all latent classes c. In other words, only the intercepts of the measurement model for the item given latent class will vary between groups.

- **N2**: `Variable: Nominal=polinter;` to specify *polinter* as nominal, and the measurement model specified in the same way as for N1. This lets $\lambda_{jlc}^{(g)} = \lambda_{jl}^{(g)}$ in (42) vary between the groups g, but in the same way for all latent classes c.

- **N3**: As N1, plus also under `Model:`

```
%class#2%
    polinter ON bul cyp;
```

  and the same for `%class#3%` and, in general, for all classes $c = 2, \ldots, C$. This lets $\lambda_{jc}^{(g)}$ in (43) vary between the classes c as well as the groups g. In other words, both the intercepts and the loadings of the measurement model for the item given latent class will vary between groups.

- **N4**: *polinter* specified as nominal as in N2, and the measurement model specified as in N3. This lets $\lambda_{jlc}^{(g)}$ in (42) vary between the classes c as well as the groups g.

Finally,

- **N5**: Like N3, but a non-equivalent measurement model specified in the same way for all three items. This is equivalent to fitting the 3-class latent class model separately for each of the groups (countries), with no parameters constrained to be equal across the groups.

We note that partial measurement models which have non-equivalence across groups are not the same for ordinal and nominal items in Mplus. Thus here N1 is equivalent with N2, and N3 is not equivalent with N4. For N5, however, ordinal and nominal models are again equivalent.

The multiple-group specification of a multigroup latent class model is rather different. First, the specification of the latent classes is changed to

```
Variable:
    Classes=country(3) class(3);
    Knownclass= country (country=1 country=2 country=4);
```

This states that there are now two latent classes, called “country” and “class”. Of these, “class” will be the actual latent class as considered above. In contrast, “country” will be a “known class”, which is specified (with the `Knownclass` command) to be exactly equal to the categories of the observed variable called *country* (the names of the observed and known class variables can but need not be the same). The multigroup model is then specified effectively as a single-group model which has these two “latent” class variables.

Here we consider only models where the items are specified as nominal. Only some of the example models in Table 5 can be estimated with the multiple-group specification (without contrived tricks through additional parameters constraints). To understand why this is done as explained below, it is useful first to understand two default settings of this specification: (i) the latent class and the "known class" are independent of each other, and (ii) the measurement model for every item is specified separately for every combination of the latent class and the known class. In other words, by default the probabilities of the latent classes do not vary between groups, but the measurement model is fully non-equivalent across the groups. To obtain other models, these defaults are changed as follows:

- **E0**: The latent class is independent of groups by default, but the measurement models for all items need to be made equivalent across groups, with

  ```
  Model class:
      %class#1%
          [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4];
      %class#2%
          [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4];
      %class#3%
          [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4];
  ```

  This requests that the intercept parameters τ_{jlc} of every items j are estimated separately for each level c of `class`, *but* independently of the "known class" variable `country`. Note that only these intercepts need to be specified here, since to Mplus the model in this specification is a single-group model and thus has no parameters $\lambda^{(g)}_{jlc}$ in (42).

- **E1**: Measurement model as in E0, plus specifying that the latent class variable is associated with the group variable, with

  ```
  Model:
      %overall%
          class ON country;
  ```

- **N4**: Like E1, expect that the `polinter#1-polinter#3` is omitted from the specification of the measurement model for each level of `class`. This implies that the measurement parameters of `polinter` will vary between the groups defined by the "known class" variable `country`.

- **N5**: The structural model (association between group and latent class) specified as in E1, but the entire `Model class` statement omitted. This leaves the measurement model in its default state of full non-equivalence across the groups.

Since it is less flexible, and typically much slower, than the covariate specification, the multiple-group specification of the multigroup latent class model is typically of limited interest. One situation where it may be a useful backup approach is where the user has access only to the demo version of Mplus. Since this places constraints on the number of explanatory variables which can be included in a model, the covariate specification is constained to a small number of groups (3, with the current demo version). The multiple-group specification avoids this limitation.

Figure 20: Mplus input syntax for a multigroup latent class model with 3 classes, in a covariate specification (Model N4X).

```
Title: LCAT_LC_N4X
    LCAT: examples of multiple-group latent variable models
    Latent class models, 3 latent classes
    Model N4
        Non-equivalence of measurement for one item, with
        interaction between group and class
        this item treated as nominal
    Covariate specification
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind bul cyp;
    Nominal = polmind polhard;
    Nominal = polinter; ! E0nom, E1nom, N2, N4, N5
!    Categorical = polinter; ! E0ord, E1ord, N1, N3
    Classes=class(3);
Define:
    cut polhard(2);
    bul = (country==2);
    cyp = (country==4);
Analysis:
    Type=Mixture;
    Estimator=ML;
Starts=50 20;
Model:
    %overall%
!        class ON bul@0 cyp@0; ! E0nom,E0ord
        class ON bul cyp; ! E1nom,E1ord,N1,N2,N3,N4,N5
        polinter ON bul cyp; ! N1,N2,N3,N4,N5
!        polmind on bul cyp; ! N5
!        polhard on bul cyp; ! N5
    %class#2%               !N3,N4,N5
        polinter ON bul cyp; ! N3,N4,N5
!        polmind on bul cyp;  ! N5
!        polhard on bul cyp;  ! N5
    %class#3%               !N3,N4,N5
        polinter ON bul cyp; !N3,N4,N5
!        polmind on bul cyp; ! N5
!        polhard on bul cyp; ! N5
Savedata:
    File="tmp.dat";
```

Figure 21: Mplus input syntax for a multigroup latent class model with 3 classes, in a multipl-group specification (Model N4MG).

```
Title: LCAT_LC_N4MG
    LCAT: examples of multiple-group latent variable models
    Latent class models, 3 latent classes
    Model N4
        Non-equivalence of measurement for one item, with
        interaction between group and class
        this item treated as nominal
    Multiple-group (known-class) specification
    Note: Commented-out lines refer to models with alternative specifications
     - see http://stats.lse.ac.uk/lcat/
Data:
    File = ess4_3c.dat;
Variable:
    Names =
        idno country ptrust pfair phelp polinter polhard polmind
        tparl tlegal tpolice tpolitic tparties;
    Missing = all(99);
    Usevariables = polinter-polmind;
    Nominal = polmind polhard polinter;
    Classes=country(3) class(3);
    Knownclass= country (country=1 country=2 country=4);
Define:
    cut polhard(2);
Analysis:
    Type=Mixture;
    Estimator=ML;
Starts=50 20;
Model:
    %overall%
        class ON country; ! E1NOM,N4,N5
Model class:
    %class#1% ! E0NOM,E1NOM,N4
!        [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4]; ! E0NOM,E1NOM
        [polhard#1 polmind#1-polmind#4]; ! N4
    %class#2% ! E0NOM,E1NOM,N4
!        [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4]; ! E0NOM,E1NOM
        [polhard#1 polmind#1-polmind#4]; ! N4
    %class#3% ! E0NOM,E1NOM,N4
!        [polhard#1 polinter#1-polinter#3 polmind#1-polmind#4]; ! E0NOM,E1NOM
        [polhard#1 polmind#1-polmind#4]; ! N4
Savedata:
    File="tmp.dat";
    Save=Cprobabilities; ! NOTE: need to include this, otherwise Mplus does not save
                         ! the known-class variable
```

5.3 Mplus output

Figure 22 shows part of the Mplus output for one of the models above (N4) fitted using the covariate specification (output for the corresponding multiple-group specification is not shown, because it would in practice be used less often; this output can be found on the LCAT website). Only parts of the output are shown, enough to explain all the different types of parameters.

Different parameters are labelled as follows in the output. The parameters of the structural model (41) are listed under `Categorical Latent Variables`:

- κ_{0c} under `Intercepts`, here $\hat{\kappa}_{01} = 1.489$ and $\hat{\kappa}_{02} = 1.620$ ($\kappa^{03} = 0$ is fixed for identification and not shown).
- $\kappa_c^{(g)}$ under `CLASS#1 ON BUL` and so on (recall that the two dummy variables for countries are called `BUL` and `CYP` here). For example, $\hat{\kappa}_1^{(2)} = -2.029$ $\hat{\kappa}_2^{(2)} = -1.075$.

These determine the probabilities of the classes. For example, here the probability that a respondent in country 2 ($g = 2$, i.e. Bulgaria) belongs to latent class 1 ($c = 1$) is

$$\hat{\alpha}_1^{(2)} = \hat{P}(\eta^{(2)} = 1) = \frac{\exp(1.489 - 2.029)}{1 + \exp(1.489 - 2.029) + \exp(1.620 - 1.075)} = 0.176.$$

The estimated parameters of the measurement models are shown separately for each latent class. Figure 22 shows them only for class 1, as the format is the same for every class. Here all the items are treated as nominal, so the measurement model is given by (42). Here we identify its parameters in the output for the item `polinter` (labelled as item $j = 1$ below) for which the measurement model is non-equivalent across the countries:

- τ_{jlc} are shown under `Intercepts` for items which are non-equivalent (as here *polinter*) and under `Means` for items which are equivalent across groups (as here *polmind* and *polhard*). For example, here $\hat{\tau}_{111} = -2.355$ and $\hat{\tau}_{121} = -0.587$.

  We note that in the Mplus output some of these measurement parameters are listed with a standard error of 0.000 and large negative (or elsewhere large positive) estimated values. These are parameters which the software fixed (rather than estimated) at such values during estimation, for reasons of numerical stability. Such values correspond to estimated item response probabilities of effectively 0 or 1.
- $\lambda_{jlc}^{(g)}$ are shown here under `POLINTER ON BUL` and so on. Under each latent class c, the estimates are shown for each group g (except for the reference group, here Belgium) within each level $l = 1, \ldots, L$ of the item (in this order; here $L = 3$ for *polinter*). Here, for example, $\hat{\lambda}_{111}^{(3)} = -1.580$ and $\hat{\lambda}_{121}^{(2)} = -4.511$.

These determine the probabilities of responses to the items given the latent classes. For example, the probability that a respondent in latent class 1 ($c = 1$) in group 2 ($g = 2$, i.e. Bulgaria) selects response level 2 ($l = 2$) for item 1 ($j = 1$, i.e. *polinter*) is

$$\begin{aligned}\pi_{12}^{(2)}(1) &= P(y_1^{(2)} = 2|\eta^{(2)} = 1) \\ &= \frac{\exp(-0.587 - 4.511)}{1 + \exp(-2.355 - 2.062) + \exp(-0.587 - 4.511) + \exp(0.141 - 1.493)} = 0.0047.\end{aligned}$$

As this example shows, calculating the latent class probabilities and response probabilities from the Mplus output is rather cumbersome and inconvenient. These probabilities are calculated automatically by the `lcat` post-pocessing functions in R (see Section 7), and displayed by them as shown in Figure 23 for the same model. A further difference between the two displays here is that in Figure 23 the latent classes have been re-numbered from Figure 22, so that classes 1, 2 and 3 in the Mplus output are classes 3, 2, and 1 respectively in the `lcat` output. Such reordering, which does not affect the fit of the model itself, is frequently done to obtain a convenient interpretation for the latent classes.

Finally, let us check the interpretation of the fitted model in this example, using the probabilities in Table 23. Recall that low values of *polinter* and *polhard*, but high values of *polmind*, correspond to high levels of political engagement:

- Class 1 has the highest probabilities for the politically engaged responses and class 3 the lowest, with class 2 in between for each of the three items. The classes thus appear to be essentially in order of decreasing engagement.
- For the item *polinter*, for which the measurement models are different in different countries, the same qualitative interpretation of the classes in relation to this item holds in each of the countries, and only the specific values of the estimated item probabilities vary across the countries.
- The estimated class probabilities indicate that the proportions of the politically engaged latent classes are highest in Cyprus and much the lowest in Belgium. This conclusion agrees with the one obtained from a latent trait model for the same items in Section 4.2.2.

Figure 22: Part of Mplus output for a multigroup latent class model with 3 classes, in a covariate specification (Model N4X).

```
                                                  Two-Tailed
                   Estimate       S.E.  Est./S.E.    P-Value

Latent Class 1

 POLINTER ON
    BUL              -2.062      0.835     -2.469      0.014
    CYP              -1.580      1.715     -0.922      0.357

 POLINTER ON
    BUL              -4.511      1.659     -2.720      0.007
    CYP             -28.756      0.000    999.000    999.000

 POLINTER ON
    BUL              -1.493      0.232     -6.450      0.000
    CYP              -1.306      0.287     -4.552      0.000

 Means
    POLMIND#1         2.690      0.282      9.544      0.000
    POLMIND#2         3.270      0.278     11.754      0.000
    POLMIND#3         1.838      0.438      4.193      0.000
    POLMIND#4       -15.000      0.000    999.000    999.000
    POLHARD#1        -2.496      0.145    -17.226      0.000

 Intercepts
    POLINTER#1       -2.355      0.276     -8.521      0.000
    POLINTER#2       -0.587      0.192     -3.058      0.002
    POLINTER#3        0.141      0.111      1.280      0.201

Latent Class 2
...


Categorical Latent Variables

 CLASS#1    ON
    BUL              -2.029      0.141    -14.346      0.000
    CYP              -2.613      0.164    -15.983      0.000

 CLASS#2    ON
    BUL              -1.075      0.140     -7.687      0.000
    CYP              -1.790      0.147    -12.183      0.000

 Intercepts
    CLASS#1           1.489      0.179      8.296      0.000
    CLASS#2           1.620      0.146     11.132      0.000
```

Figure 23: Output from the `lcat` post-processing functions in R for a multigroup latent class model with 3 classes fitted in Mplus (Model N4X). Mplus output for the same model is shown in Figure 22. The latent classes labelled 1, 2, 3 there are labelled 3, 2, 1 here respectively.

```
--------------------------------------------------------------------------------
LCAT output
Mplus file:  lcat_lc_n4x
Latent class model, latent class variable CLASS with 3 classes

3 categorical items:
 Name     Level   Categories Invariant
 POLMIND  Nominal 5          Yes
 POLHARD  Nominal 2          Yes
 POLINTER Nominal 4          No
Multiple group model, with 3 groups:
BEL  BUL  CYP

Model estimates:
N = 5200        parameters = 48                 log-likelihood = -15821.5
AIC = 31739     BIC = 32053.71  Delta =  0.12 ( 0.11 - 0.129 across groups )
% of 2-way marginal residuals > 4: 29 ( 13 - 34 across groups )

Probabilities of latent classes:
                        CLASS#1 CLASS#2 CLASS#3
BEL                     0.095   0.482   0.423
BUL                     0.302   0.521   0.176
CYP                     0.461   0.389   0.150

Measurement probabilities:
For items that are invariant across groups:
                        CLASS#1 CLASS#2 CLASS#3
        POLMIND#1       0.006   0.000   0.305
        POLMIND#2       0.000   0.204   0.544
        POLMIND#3       0.023   0.583   0.130
        POLMIND#4       0.633   0.213   0.000
        POLMIND#5       0.338   0.000   0.021

        POLHARD#1       0.817   0.207   0.076
        POLHARD#2       0.183   0.793   0.924

For items that are not invariant across groups:
                        CLASS#1 CLASS#2 CLASS#3
BEL     POLINTER#1      0.376   0.068   0.034
BUL     POLINTER#1      0.252   0.051   0.009
CYP     POLINTER#1      0.358   0.046   0.015

BEL     POLINTER#2      0.449   0.570   0.198
BUL     POLINTER#2      0.466   0.429   0.005
CYP     POLINTER#2      0.375   0.236   0.000

BEL     POLINTER#3      0.086   0.294   0.411
BUL     POLINTER#3      0.158   0.381   0.203
CYP     POLINTER#3      0.181   0.520   0.234

BEL     POLINTER#4      0.089   0.068   0.357
BUL     POLINTER#4      0.124   0.138   0.783
CYP     POLINTER#4      0.086   0.198   0.751
--------------------------------------------------------------------------------
```

6 Other topics

In this section we briefly comment on two major issues related to latent variable modelling which are not covered in detail in this document: (i) numerical implementation of the estimation of the models in Mplus, and (ii) model assessment statistics included in the output.

6.1 Estimation of the models

For the latent trait and latent class models discussed in Sections 4 and 5 we have assumed that the parameters of the models are estimated using maximum likelihood (ML) estimation. This requires the use of iterative numerical algorithms, and appropriate choices for their settings, including the following:

- Choice of the algorithm (or a combination of algorithms) itself.
- Selection of starting values. For the models considered here, the log-likelihood function typically has multiple local maxima, so it is almost always necessary to use multiple starting values to increase the chance that the global maximum is found.
- Convergence criterion and stopping rule for the algorithm.
- For latent trait models, the method of numerical integration used to evaluate the value of the log-likelihood function.

In Mplus, all of these are specified by options of the `ANALYSIS` command. Please see the Mplus user's guide (Muthén and Muthén 2007) and technical appendices (Muthén 2004) for information on the choices that are implemented in the programme.

6.2 Model assessment statistics

Various model assessment statistics are included in the Mplus output or can be derived from it. Here we list briefly some of them, focusing on statistics that are used for latent class and latent trait models. For factor analysis models, a still longer list of statistics is conventionally provided, but they are not discussed here. We list only those statistics which are included in the output of the `lcat` post-processing functions (see Section 7 and Appendix C.2):

- Sample size (here denoted n).
- Number of estimable parameters in the model (r).
- Log-likelihood of the fitted model ($\log L$).
- Akaike's information criterion: $AIC = -2 \log L + 2r$.
- Bayesian information criterion: $BIC = -2 \log L + (\log n)r$.

- Index of dissimilarity ("Delta")

$$\widehat{\Delta} = \frac{\sum_{k=1}^{K} |O_k - E_k|}{2n}$$

  where the sum is over all the K cells of the $(p+1)$-way contingency table of p items by group, and O_k and E_k denote the observed and expected (fitted) counts in cell k respectively. The `lcat` functions also show the values of $\widehat{\Delta}$ separately for each group, i.e. calculated for the p-way contingency table of the items within each group.

The `lcat` functions also show statistics based on standardised two-way marginal residuals. To define these, consider the two-way table of frequencies for items i and j (or one item and the group), after collapsing the full table over all the other variables. Let $O_{st}^{(ij)}$ denote the observed count in this table for the cell which corresponds to levels s and t of i and j respectively, and let $E_{st}^{(ij)}$ denote the corresponding expected (fitted) count obtained by collapsing the full table of expected counts similarly. The standardised bivariate marginal residuals are defined by

$$R_{st}^{(ij)} = \frac{(O_{st}^{(ij)} - E_{st}^{(ij)})^2}{E_{st}^{(ij)}}$$

for each combination of values s, t of each pair of variables i, j. For a rough rule of thumb, we may treat a value of $R_{st}^{(ij)}$ greater than 4 as tentative evidence of poor fit. The `lcat` residuals function (see Section 7.8 and Appendix C.2) shows various summaries of $R_{st}^{(ij)}$, such as (i) all of their individual values, and those that are greater than 4; (ii) the percentage of $R_{st}^{(ij)}$ which are greater than 4, both overall and separately for each group; (ii) sums $S^{(ij)}$ of $R_{st}^{(ij)}$ over the levels s, t for each pair (i, j) and, when i and j are both items, this sum separately for each group.

7 Using R software to work with Mplus

A set of functions in the R language has been written for the LCAT project to facilitate multi-group latent variable modelling with Mplus. These functions have three main roles:

- Importing results of a fitted model from Mplus into an R object.
- Calculating bivariate marginal residuals for models for categorical responses (i.e. latent trait and latent class models).
- Facilitating further presentation of the model results, such as tables of estimated parameters and residuals and plots of fitted probabilities.

Main features of the functions are summarised below, and their syntax is listed in Appendix C. The functions use the add-on package `MplusAutomation` written by Michael Hallquist. This is explained in Appendix B.

7.1 Installation

The following need to be done once:

- Install an up-to-date version of R (at the time of writing 2.15.0) from `http://cran.r-project.org/bin/windows/base/` by downloading and then executing the .exe file accessed through a link on this page.
- In R, install the `MplusAutomation` package with `install.packages("MplusAutomation")`.
  - In every R session `MplusAutomation` will then need to be loaded by `library(MplusAutomation)`. This is done automatically by the `lcat` function.
- Load `lcat` and related functions to the R workspace. The current way of doing that is explained on the LCAT website, at `stats.lse.ac.uk/lcat/`.

7.2 Basic ideas of the `lcat` functions

In summary, `lcat` and related functions are used as follows

- `lcat` itself executes Mplus input (this part is optional) and reads in the resulting output. The results are then processed in R and saved in an object; for example,
  ```
  trustmodels<-lcat("trust1cl.out",path="c:/lcat")
  ```
  The resulting object is a list of class `lcat.list` with the following elements:
  - `summary`: a summary of the models on the list
  - `results`: a list in which each element is an object of class `lcat`, containing results for a single model
  - `lr.tests`: a table of results of likelihood ratio tests between models on the list (this is initially empty, and created subsequently by calls to the `lcat.lrtest` function).

- `print.lcat.list` prints formatted summary tables of the whole list. This can be invoked simply by typing the name of the list; for example,
  `trustmodels` or `print(trustmodels)`

- `print.lcat` prints formatted summary tables of individual models on the list; for example,
  `print(trustmodels,2)`

- `reorder.lcat.list` reorders or relabels various elements of the models and their results; for example
  `trusmodels<-reorder(trustmodels,1,classes=c(3,1,2)))`
  Note that the function is "generic" in the R terminology. This is why we call it as just `reorder`; when the first argument of the function is an object of class `lcat.list`, as produced by the `lcat` function, the call to `reorder` automatically invokes the function `reorder.lcat.list`. The same applies to the functions below.

- `plot.lcat.list` Draws plots of items response probabilities for latent trait and latent class models; for example,
  `plot(trustmodels,models=1:2,items=1,levels=1)`

- `lcat.lrtest` calculates likelihood ratio tests between models on the list; for example,
  `lcat.lrtest(trustmodels,1,2)`
  The results of the test are automatically added to the `lr.tests` object of the list.

- `residuals.lcat.list` (which can be shortened to `resid`) prints selected residuals for one of the models on the list; for example
  `resid(trustmodels,1,item2way=T,over4=T,group="BEL")`

7.3 Uploading models: The `lcat` main function

7.3.1 Setup

- Suppose you are running R in a workspace, where you have access to the LCAT functions.

- You have Mplus input files and data in some other directory. For example, suppose that this is `c:/lcat/models`, and that it includes input files `model1.inp`, `model2.inp` and `oldmodel1.inp`. If these models have already been fitted in Mplus, it will also contain the corresponding output files `model1.out`, `model2.out` and `oldmodel1.out`.

  - The input file must include a `SAVEDATA: File=<filename>;` command. The name `<filename>` can be the same for several input files, but only if they are then uploaded to R one model at the time, so that the saved data file always corresponds to the model which is being uploaded.
  - For latent class and latent trait models which are fitted in a way which involves a `Knownclass` variable, the input file should also include a `Save=Cprobabilities;` option under the `SAVEDATA` command.

7.3.2 Referring to paths and files

Consider the example defined above. The first two arguments of the `lcat` function are called `target` and `path`. They are combined in various ways to upload some or all of the output files in a directory:

1. All the files in the directory: `models<-lcat(target=NULL, path="c:/lcat/models")` (NOTE: No backslash at the end)

2. Some files in the directory:
   `models<-lcat(target=NULL,path="c:/lcat/models",filefilter="model.*")`.
   Here the `.*` is a wildcard which stands for "any number of any characters". In this example, `oldmodel1` would be excluded, whereas `filefilter=".*model.*` would include it. Note also that the extension `.out` is not considered, so `filefilter=".*model1"` would find both `model1.out` and `oldmodel1.out`.

3. One file in the directory:
   `models<-lcat(target="c:/lcat/models/model1.out")` or
   `models<-lcat(target="model1.out",path="c:/lcat/models")`

Note that you can omit the `target=` and `path=` in each of these, as long as they are listed as first and second arguments in this order.

Note that these need to be modified if input files are also run with the `runmodels=T` option (see below): 1. works as before; 2. does not work; and 3. works, but it causes as the input files in the target directory to be run, even if only the one one is then loaded into R.

Further selection of which models are run and/or loaded is implemented by the `overwrite` and `replaceOutfile` options, which are explained below.

7.3.3 Running models from R

The `runmodels=T` option of `lcat` causes the input files in the target directory to be run first, before the output is loaded. Which input files are run is controlled by the `replaceOutfile` option:

- if `replaceOutfile="never"`, only those input files for which the `.out` files does not exist are run
- if `replaceOutfile="always"`, all input files are run, overwriting any existing `.out` files

7.3.4 Loading results into R

Typically the results read by `lcat` are assigned to an object (which will be a list of class `lcat.list`). Subsequent calls to `lcat` can be used to add models to the same list, using the addto option. If a model with the same name exists on the list already, the overwrite option controls what happens. Examples:

- models<-lcat(NULL,"c:/lcat/models") loads results for all three models in the directory and saves them in the object models. The models in models are names with the names of the output files (without the .out), i.e. here model1, model2 and oldmodel1.
- models<-lcat("model2.out","c:/lcat/models",addto=models) tries to load results from output file model2.out and add them to the list models, resaving the result as model. This works if models does not yet include an element with the name model2; if it does, this call does nothing and produces a warning message (because the default value of overwrite is FALSE).
- models<-lcat("model2.out","c:/lcat/models",addto=models,overwrite=T) loads results from output file model2.out and adds them to the list models, resaving the result as model. This overwrites the element model2, if it already exists. Any likelihood ratio tests involving this model in the lr.tests element of models are also replaced.

7.4 Looking at the results: print

Typing just the name of a model list, e.g.
models
prints tables of summary statistics (and likelihood ratio tests, if present) for all the models on the list models (this is short for print(models), which is alias for print.lcat.list(models)).

print(models,1) prints a summary of model 1 on the list models.

- The same can be obtained with models[[2]][[1]] or models$results[[1]], all of which ultimately call print.lcat.
- If you want to see the contents of an lcat object without formatting, try print.default(models$results[[1]]) (warning: this prints quite a lot of output).
- print(models,1,alt=T) prints a different presentation of the fitted model, depending on the type of model.
- print(models,1,Mplus=T) prints a table of the parameter estimates extracted from Mplus output, without further formatting. This is useful for finding the standard errors of estimates, which are not included in lcat output.
- print(models,1,allMplus=T) prints a copy of the entire Mplus output file.

7.5 Tidying up the results: reorder

The function reorder.lcat.list is used to tidy up models in a lcat.list object in various ways. The function automatically modifies the list that it is called on. It also returns invisible the modified list, which can thus also be assigned to a different name.

The function can do the following things:

- Reordering and/or deleting models on the list: Suppose that here the list `models` has three elements. The numbers of the elements can be seen by typing the name of the list, here `models`.
  - `reorder(models,models=c(3,2,1))`. This means that the old element 3 becomes new 1, 2 becomes 2 and 1 becomes 3. Note that here the first `models` is the name of the list, which can be anything, but the second `models` is the name of an argument of `reorder.lcat.list`.
  - `reorder(models,models=c(2,1))`. This means that the order of old elements 1 and 2 is reversed, and element 3 is deleted.
- Reordering and/or deleting likelihood ratio tests:
  - `reorder(models,tests=c(2,1))`. This changes the order of tests in the `lr.tests` element of `models`, with the same logic as with the `models` option. Tests can again be deleted as well. The numbers are as listed in the table of the tests when you type the name of the list (here `models`).
- Reordering groups: Suppose that models in `models` are multiple-group models, each with 3 groups. The current order of the groups can be seen in the summary of the models.
  - `reorder(models, groups=c(3,1,2), elements="All")`. This reorders the groups. The option `elements="All"` means that the same reordering is applied to all models in the list. This is the default (so it can be left out).
- Relabelling groups:
  - `reorder(models, groupnames=c("England","Scotland","Wales"))`. This replaces the previous *labels* of the groups in a multiple-group model.
- Relabelling the reference group: When a multiple-group model is fitted using a dummy-variable specification, the reference group (i.e. the one without a dummy variable) is labelled "Ref.group" by default. This is relabelled by
  - `reorder(models, refname="England")`
- Reordering latent classes: This can only be done one model at a time. Suppose model 2 in `models` is a latent class model with 3 classes. The order of these is changed by
  - `reorder(models, classes=c(2,3,1))`
- Reordering latent classes for one group only: This makes sense *only* for a multiple-group model with complete measurement nonequivalence across groups.
  - `reorder(models, classes=c(2,3,1), only.group="Scotland")`
- Reordering and/or reversing latent traits:
  - `reorder(models, traits=-1, elements=2)`. Here element 2 of `models` is a 1-trait latent trait model. The command reverses that trait.

- `reorder(models, traits=c(2,-1), elements=3)`. Here element 3 of `models` is a 2-trait latent trait model. The command makes old trait 2 new trait 1 and old 1 the new 2, and reverses the new trait 2 (old 1).
- This can also be used with the `only.group` option, as for latent class models. This again makes sense only for a model with complete nonequivalence of measurement.

- Reordering levels of an observed indicator variable:
  - `reorder(models,"All",ylevels=list("item1",3:1))`. This changes the order of categories of item called `item1` in all models of list `models`. Old levels 3, 2, 1 become new levels 1, 2, 3.

7.6 Plots of item response probabilities: `plot`

The function `plot.lcat.list` draws plots of item response probabilities for both latent trait and latent class models. The basic call for the function is of the form

```
plot(models,models=<models>,items=<items>,levels=<levels>,groups=<groups>)
```

where `mymodels` is a name of an lcat list object, and each of `<models>`, `<items>`, `<levels>` and `<groups>` is a vector of numbers, of models (in the lcat list), items (observed indicator variables), levels (of the indicators) and groups (in a multigroup model) respectively. For `models` and `levels`, the default value is 1. The numbering refers to the order in which each of these are listed in `print.lcat.list` or `print.lcat` output. Response probability curves are plotted for each combination of the options; for example,

- `plot(mymodels,items=1,levels=1:2)`: Levels 1 and 2 of item 1 in group 1 in model 1.
- `plot(mymodels,items=1:2,levels=2,models=c(1,3))`: Level 2 of items 1 and 2 in group 1 in each of models 1 and 3 (i.e. 4 curves in total).
- `plot(mymodels,items=3,levels=2,groups=c(2,1))`: Level 2 of item 3 in model 1, for groups 2 and 1. Here (and in other options) `groups=c(2,1)` differs from `groups=c(1,2)` in the order in which the curves are listed in the legend of the plot.

The option `plotlist` is an alternative to this specification. It can be used to request any set of plots. This argument is a list of vectors, each of which is of length 3, 4 or 5, most commonly 3 or 4. If 5, the elements of the vector are model, item, level, group and reference group; if 4, reference group is inferred from the input; if 3, group is assumed to be 1. For example:

- `plot(mymodels,plotlist=list(c(1,1,1),c(1,2,1),c(2,1,1),c(2,2,1)))` is the same as `plot(mymodels,models=1:2,items=1:2,levels=1)`.
- `plot(mymodels,plotlist=list(c(1,3,4,1),c(2,3,4,1),c(2,3,4,5)))` plots the curves for level 4 of item 3 for group 1 in model 1, and groups 1 and 5 in model 2. This plot can only be produced with the `plotlist` option.

The `cumprob` option is used to plot cumulative rather than individual category probabilties. Its values are FALSE (the default, individual probabilities are plotted), "low" (sum of probabilities from level 1 up and including a given level) and "high" (sum of probabilities starting from a given level and up to the last one). For example:

- `plot(mymodels,items=5,levels=2,cumprob="low")`. For model 1 and group 1, probabilities that item 5 has level 1 or 2.
- `plot(mymodels,items=5,levels=2,cumprob="high")`. For model 1 and group 1, probabilities that item 5 has level 2 or any higher-numbered level.

The option `trait` specifies the latent trait which is varied in a plot for a latent trait model. It must be a single number. By default, `trait=1`.

The function also has several other options, mainly for modifying the appearance of the plot. These are listed in Appendix C.

7.7 Likelihood ratio tests: `lcat.lrtest`

The function `lcat.lrtest` carries out likelihood ratio tests between elements of an `lcat.list` object. For example,

```
lcat.lrtest(models,1,2)
```

carries out the test between elements 1 and 2 of `models` (the numbers are listed next to the model results when you type `models`.) The results of the test are then added to the `lr.tests` element of `models`.

The function carries out only the most rudimentary checks that the test for the pair of models is actually appropriate. This is mostly the user's responsibility.

Note that tests can be carried out also between results of (appropriately nested) models fitted using the "known-class" and "covariate" specifications in Mplus. This is because `lcat` removes from the log-likelihood the component which corresponds to the marginal distribution of the explanatory variable (group) for models fitted with the known-class specification, where it is included by Mplus. The same is done for AIC, BIC and the number of parameters.

7.8 Residuals: `resid`

Tables of residuals are obtained with the `residuals.lcat.list` function, which can be invoked by calling `resid`. This has various options. Examples of its use:

- `resid(models,1,full=T)`: individual cell residuals for the full table of groups and items, for model 1 in `models`
- `resid(models,1,item2way=T, over4=T)`: residuals for marginal 2-way item-by-item tables (conditional on group, and unconditional), showing only the ones where the standardised residual ($(O-E)^2/E$) is over 4 in absolute value. The option `item2way=T` is the default, so these residuals are shown unless otherwise specified.

- `resid(models,1,group2way=T, sort=T)`: residuals for marginal 2-way group-by-item tables, sorted in descending order of the standardised residuals
- `resid(models,1,group="UK", item="trust1")`: residuals for marginal 2-way item-by-item tables, conditional on group "UK", and showing only residuals involving item "item1"
- `resid(models,1,item=c("trust1","trust2"))`: residuals for marginal 2-way item-by-item tables, showing only residuals between items "item1" and "item2".
- `resid(models,1,sumitem2way=T)`: sums of 2-way item-by-item residuals for each pair of items, conditionally on group and unconditionally.
- `resid(models,1,sumgroup2way=T)`: sums of 2-way group-by-item residuals for each item.

A Data set used for the examples

The data are from Round 4 of the European Social Survey (ESS)[4]. The variables are as listed below, with concise descriptions. The first label is the variable name we have used within Mplus. In brackets, the number of levels of the variable, its name in the ESS SPSS file and question number in the ESS questionnaire.

- `idno`: person ID
- `country` of the respondent (1=Belgium, 2=Bulgaria, 4=Cyprus; `cntrynum`)
- Indicators of interpersonal trust:
  - `ptrust`: "Most people can be trusted or you can't be too careful" (0–10; `ppltrst`, A8)
  - `pfair`: "Most people try to take advantage of you, or try to be fair" (0–10; `pplfair`, A9)
  - `phelp`: "Most of the time people helpful or mostly looking out for themselves" (0–10; `pplhlp`, A10)
- Indicators of interest in politics:
  - `polinter`: "How interested in politics" (1–4; `polintr`, B1)
  - `polhard`: "Politics too complicated to understand" (1–5; `polcmpl`, B2)
  - `polmind`: "Making mind up about political issues [how hard]" (1–5; `poldcs`, B3)
- Indicators of institutional trust:
  - `tparl`: Trust in country's parliament (0–10; `trstprl`, B4)
  - `tlegal`: Trust in the legal system (0–10; `trstlgl`, B5)
  - `tpolice`: Trust in the police (0–10; `trstplc`, B6)
  - `tpolitic`: Trust in politicians (0–10; `trstplt`, B7)
  - `tparties`: Trust in political parties (0–10; `trstprt`, B8)

The values are coded so that larger values indicate higher levels of trust and better interest/understanding of politics, with the exception of `polhard` for which the reverse is true.

There are no missing values of country. Total sample size is 5205, with 1760, 2230 and 1215 for Belgium, Bulgaria and Cyprus respectively. Individual variables have missing values. In the SPSS data file obtained from the ESS website values in the range of 6–9 are missing-value codes for `polintr`, `polcmpl` and `poldcs`, and values 66–99 for the other variables. In our analyses all missing-value codes have been combined into one.

The Mplus syntax files shown in this manual assume that the data have been saved in a text file called `ess4_3c.dat`. See Section 1.2.1 for how to create such a file from SPSS, Stata and R. The syntax examples assume that all missing values are coded in the data set as 99. See the examples in Section 1.2.1 for modifications of the syntax in other cases.

[4]ESS Round 4: European Social Survey Round 4 Data (2008). Data file edition 4.0. Norwegian Social Science Data Services, Norway — Data Archive and distributor of ESS data. The data were downloaded from `http://ess.nsd.uib.no/`

B MplusAutomation package in R

The `MplusAutomation` (see `http://cran.r-project.org/web/packages/MplusAutomation/`) add-on package in R facilitates processing input to and output from Mplus. The use of the package is briefly outlined here. Further information can be found in the help files (enter `help(MplusAutomation)`) and examples of use in a vignette document `http://cran.r-project.org/web/packages/ MplusAutomation/vignettes/Vignette.pdf`.

We use `MplusAutomation` through the `lcat` function, explained in Appendix 7, so the information in this appendix is not typically needed for daily use. It is provided here for completeness. Below, we abbreviate `MplusAutomation` as `MpA`.

The `MpA` package has three primary purposes:

1. To create data files for input into Mplus (function `prepareMplusData`) and related input files (`createModels`).
2. To automatically run groups/batches of models (`runModels`, `runModels_Interactive`).
3. To provide routines to extract model fit statistics, parameter estimates, and raw data from Mplus output files (`readModels`, `extractModelParameters`, `extractModelSummaries`, `getSavedata_Data`, `getSavedata_Fileinfo`, `showSummaryTable`, `LatexSummaryTable`, `HTMLSummaryTable`)

B.1 Creating input data and command files

(1) `prepareMplusData`: This simply creates from an R data frame a tab-limited text file which is suitable for reading into Mplus. There are options for specifying which variables in the data frame are included in the output file. Example:

```
prepareMplusData(ess4.3c.dat,"d:/lcat/ess4_3c.dat")
```

Here `ess4.3c.dat` is an R data frame. The command also prints on the R console commands for reading in the data, which can then be copied into an Mplus command file. In this example these commands are

```
TITLE: Your title goes here
DATA: FILE = "d:/lcat/ess4\_3c.dat";
VARIABLE: NAMES = idno ptrust pfair phelp polinter polhard polmind tparl tlegal
     tpolice tpolitic tparties country;
MISSING=.;
```

(2) `createModels(templatefile)`: This creates a set of Mplus input files which differ systematically in some specification, as specified in the text file `templatefile`. A simple example is specification of latent class models with 2–7 classes, which requires 6 input files which differ only in one place. Much more complex uses are possible. The vignettes document gives details of the structure of the template files. LCAT applications will be developed if needed.

B.2 Running models in Mplus

runModels: This runs Mplus for all models in a given directory (and its subdirectories, if requested), or all input files for which the directory does not contain a corresponding output (.out) file. An example:

```
runModels("d:/lcat",replaceOutfile="always",showOutput=TRUE)
```

Here replaceOutfile="always" runs all input files in the directory, and replaceOutfile="never" only those without an .out file. The option showOutput=TRUE prints the Mplus estimation output (on progress of the iterations) on the R console; the default of this option is FALSE.

runModels_Interactive: This provides a graphical interface for selecting options for and then running runModels.

B.3 Returning Mplus results to R

readModels: This reads parameter estimates, summary statistics and (if saved with the Mplus SAVEDATA command) analysis data set for fitted models in a directiry into an R object. Example:

```
modelresults<-readModels("d:/lcat")
```

There is an option filefilter for specifying which output files from the directory should be included (this uses the powerful but elaborate syntax of "regular expressions", see http://www.regular-expressions.info/).

readModels is a wrapper for functions extractModelParameters, extractModelSummaries and getSavedata_Data which extract only parameter estimates, summary statistics and saved data respectively (there is also getSavedata_Fileinfo which gets only information about the saved data file, especially the variable names). These can also be called individually. Each of them allows the argument to be a single output file rather than a whole directory, as in

```
extractModelSummaries("d:/lcat/lcatm.3.1a.out")
```

showSummaryTable, LatexSummaryTable, and HTMLSummaryTable: These functions redisplay model summaries as a table in a separate window, LaTeXtable (printed on R console) and HTML file. We will mostly use the functions discussed in Section 7 for this purpose.

C Syntax of lcat functions

C.1 Main function lcat

```
lcat <- function (target, path = NULL, recursive = FALSE, filefilter,
    addto = NULL, overwrite = FALSE, print = FALSE, runmodels = FALSE,
    replaceOutfile = "never", nsimLattrait = 0, ...)

Arguments:
Arguments that are passed to MplusAutomation functions:

target, path
                Strings, which may be NULL. These are combined to form
                the directory containing Mplus output files (.out) to parse
                OR the single output file to be parsed. May be a full path,
                relative path, or a filename within the working directory.
                Example: target="model1.out", path="C:/Mplus_Runs"
recursive       If TRUE, parse all models nested in subdirectories
                within target. Defaults to FALSE.
filefilter      a Perl regular expression (PCRE-compatible) specifying particular
                output files to be parsed within directory.
                See regex or http://www.pcre.org/pcre.txt for details about
                regular expression syntax.
replaceOutfile  if runmodels=TRUE, which input files will be run.
                Currently supports three settings: "always", which runs
                all models, regardless of whether an output file
                for the model exists; "never", which does not run any model
                that has an existing output file;
                and "modifiedDate", which only runs a model if the modified
                date for the input file is more recent than the output file
                modified date (implying there have been updates to the model).

Other arguments:
addto           existing list of the class lcat.list, to which new
                results will be appended (or existing ones replaced,
                if addto=TRUE)

overwrite       logical. If TRUE, existing results with the same output filename
                are replaced. If FALSE, old results are not replaced
                and a warning message is given

print           logical. If TRUE, basic output of models read in is
                printed on the screen

runmodels       logical. If TRUE, input files in the directory are run,
                as specified by replaceOutfile

nsimLattrait    numerical scalar, which determines how integration
                required for for fitted values for latent trait models
                is carried out. If 0, numerical integration (with the cubature
                package) is used. If >0, number of draws used for Monte Carlo
                integration.
---------------------------------------------------------------------------
Return value:   a list of class lcat.list
```

C.2 Residuals function resid (i.e. residuals.lcat)

```
residuals.lcat.list <- function (x, num, item2way = TRUE, over4 = FALSE, full = FALSE,
    group2way = FALSE, group = NULL, item = NULL, sort = FALSE,
    sumitem2way = FALSE, sumgroup2way = FALSE)

Arguments:

x               name of an lcat.list object

num             number of the model in x for which residuals are requested
                if 0, a warning message is produced

The rest of the arguments of indicate which statistics are requested:

item2way        logical. If TRUE, marginal residuals for 2-way
                item-by-item tables are printed

over4           logical. If TRUE, only standardised residuals ((O-E)^2/E)
                greater than 4 in absolute value are printed

full            logical. If TRUE, cell residuals for the full table of
                items and groups are printed

group2way       logical. If TRUE, marginal residuals for 2-way
                group-by-item tables are printed

group           character. If not NULL, results are printed only for
                the group whose name matches group

item            character. If not NULL, results are printed only for
                the item whose name matches group

sort            logical. If TRUE, results are printed in descending order
                of absolute values of standardised residuals ((O-E)^2/E)

sumitem2way     logical. If TRUE, the sums of 2-way item-by-item
                residuals, summed over categories of the items, are
                returned for each pair of items, for the whole sample and
                conditional on each group

sumgroup2way    logical. If TRUE, the sum of all 2-way item-by-group
                residuals, summed over the groups, is returned for each
                item

group and item do not currently work with full=TRUE

---------------------------------------------------------------------------
Return value:

A data frame, matrix or list of the selected rasiduals.
```

C.3 Likelihood ratio test function `lcat.lrtest`

```
lcat.lrtest <- function (x, first, second, print = TRUE, update = TRUE)


Arguments:

x               name of an lcat.list object

first, second   numerical scalars. A likelihood test is carried out
                between elements first and second of x
                Note: the function does only very rudimentary tests
                that the models are actually nested and appropriate
                for testing with the LR test. This is mostly the
                responsibility of the user

print           logical. If TRUE, the results are also printed on the
                console

update          logical. If TRUE, the results of the test are added to the
                lr.tests element of x. If FALSE, the results are
                returned as a data.frame

---------------------------------------------------------------------------
Return value:

If update=TRUE, a change to the lr.tests element of the lcat.list object x.
If update=FALSE, a data frame with the results of the test.
```

C.4 Post-processing function `reorder` (i.e. `reorder.lcat.list`)

```
reorder.lcat.list <- function (lcat.list, elements="All",
        models=seq(length(lcat.list[[2]])),
        tests=NULL, groups=NULL, classes=NULL, traits=NULL,
        only.group=NULL, ylevels=NULL, refname=NULL, groupnames=NULL, update=TRUE)

Aguments:

lcat.list       name of an lcat.list object

elements        numerical scalar or "All". If numerical, only the
                element of lcat.list given by elements is affected;
                if "All", change is applied to all elements. "All" is
                not allowed with classes, and only "All" is allowed
                with ylevels

models          numerical vector. Elements of lcat.list are reordered
                according to models. For example, if models=c(3,2,1),
                element 3 of lcat.list becomes element 1 of the new
                list, and so on. Elements of lcat.list which are not
```

```
                mentioned in models are deleted.
                This is ignored if any of the other options below are
                not NULL.

tests           numerical vector. This reorders and/or deletes rows of
                the lr.tests element of lcat.list, in the same way as
                models does for the models.

[The arguments groups, traits, classes and ylevels[[2]] are numerical
reordering vectors as models and tests, except that deleting values
is not allowed.]

groups          If not NULL, order of groups is changed

classes         If not NULL, order of latent classes is changed.

traits          If not NULL, order of latent traits is changed. Negative
                elements in traits imply that the direction of a trait
                is changed.

only.group      string scalar, used in conjunction with classes. If not NULL,
                ordering of latent classes in group only.group only is
                changed, according to classes. This only makes sense if
                the model is a separate-groups latent class model, i.e.
                with complete measurement non-equivalence across groups.

ylevels         a list of length 2, with elements (i) name of a response
                variable, and (ii) a reordering vector. If not NULL,
                this changes the order of the categories of the named item

refname         string scalar. Used with multiple-group models fitted using the
                dummy variable specification. Changes the name of the reference
                group from the default "Ref.group" to refname

groupnames      string vector. If not NULL, changes the names of the
                groups in a multiple-group model.

update          logical scalar. If TRUE, the lcat.list object with the
                changes implemented by the call to the function is
                returned under the name of the object on which the
                function was called, and replaces that object
---------------------------------------------------------------------------
Return value:   a list of class lcat.list
```

C.5 Printing function `print` (i.e. `print.lcat.list`)

```
print.lcat.list <- function (x, num = 0, round = 3, alt = F, Mplus = F, allMplus = F)

Arguments:

x               name of an lcat.list object

num             numerical scalar. If 0, a summary of all the models
                in the object is printed. If >0, a summary of the
                (num)th model on the list is printed

round           numerical scalar. The results are printed rounded to
                this many decimal places.

[The following arguments work only if num>0]

alt             logical scalar. Relevant only for latent trait and
                factor analysis models. If TRUE, prints a different
                version of the output table. For latent trait models,
                this contains the parameters of the measurement models
                instead of fitted probabilities. For factor analysis
                models, this contains fitted values of the items at different
                values of the factors, instead of measurement
                parameters. For both, the parameters of the structural
                model (means and standard deviations of latent
                variables) are shown in the for of differences from
                a reference group, instead of the estimated parameters
                for each group.

Mplus           logical scalar. If TRUE, prints a concise, unformatted
                matrix of parameter estimates and their standard errors
                and p-values directly from Mplus.

allMplus        logical scalar. If TRUE, prints the entire Mplus
                output, completely unformatted.
---------------------------------------------------------------------------
Return value:   NULL. Results are printed on the screen.
```

C.6 Plotting function plot (i.e. plot.lcat.list)

```
print.lcat.list <- function (lcat.list, items, levels, models = 1, groups = 1,
    refgroups = NULL, plotlist = NULL, trait = 1, range = NULL, cumprob = FALSE,
    lty = 1:5, lwd = 2, col = 1:6, use.col = TRUE, pch = 0:18,
    xlab = NULL, ylab = NULL, plot.legend = T, legend.txt = NULL,
    xl = NULL, yl = NULL, ...)

Arguments:
lcat.list       name of an lcat.list object

items           numerical vector. Items for which probabilities are
                plotted.

levels          numerical vector. Levels of the items for which
                probabilities are plotted.

models          numerical vector. Elements of lcat.list for which
                the requested probabilities are plotted.

groups          numerical vector. Groups for which
                probabilities are plotted.

refgroups       numerical vector. Used together with plotlist to define
                reference groups for the plots. Usually NULL.
plotlist        a list of vectors. An alternative to specifying the
                plot using the arguments above. It can be used to request any
                set of plots. This argument is a list of vectors, each of which
                is of length 3, 4 or 5, most commonly 3 or 4. If 5, the elements
                of the vector are model, item, level, group and reference group;
                if 4, reference group is inferred from the input;
                if 3, group is assumed to be 1. For example:
          plot(mymodels,plotlist=list(c(1,1,1),c(1,2,1),c(2,1,1),c(2,2,1)))
                is the same as plot(mymodels,models=1:2,items=1:2,levels=1)}
          plot(mymodels,plotlist=list(c(1,3,4,1),c(2,3,4,1),c(2,3,4,5)))}
                plots the curves for level 4 of item 3 for group 1 in model 1,
                and groups 1 and 5 in model 2. This plot can only be produced
                with the plotlist option.

trait           numerical scalar. Specifies the trait which is varied
                in a plot of probabilities for a latent trait model.

range           numerical vector of length 2. The range of values over which
                probabilities given a continuous latent trait are plotted.
                This cannot be wider than (-2,2).

cumprob         FALSE, "low" or "high". If FALSE, individual category
                probabilities are plotted. If "low", cumulative
                probabilities starting from the lowest-numbered category
                are plotted. If "high", cumulative probabilities
                starting from the highest-numbered category are plotted.
Other arguments: These affect the appearance of the plot.
---------------------------------------------------------------------------
Return value:   NULL. Results are printed on the screen.
```